Relating Proof Systems for Recursive Types

ISBN 90-9019086-4

Typeset with LaTeX2e
Cover design by Michel Klein and Clemens Grabmayer
Printed by PrintPartners Ipskamp, Enschede, the Netherlands

Author's address:
Department of Computer Science
Vrije Universiteit Amsterdam
De Boelelaan 1081a
1081 HV Amsterdam
The Netherlands
`clemens@cs.vu.nl`
`http://www.cs.vu.nl/~clemens/`

VRIJE UNIVERSITEIT

Relating Proof Systems for Recursive Types

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad van doctor aan
de Vrije Universiteit Amsterdam,
op gezag van de rector magnificus
prof.dr. T. Sminia,
in het openbaar te verdedigen
ten overstaan van de promotiecommissie
van de faculteit der Exacte Wetenschappen
op dinsdag 22 maart 2005 om 13.45 uur
in de aula van de universiteit,
De Boelelaan 1105

door

Clemens Armin Grabmayer

geboren te Linz, Oostenrijk

promotor: prof.dr. J.W. Klop
copromotor: dr. R.C. de Vrijer

Contents

Preface

At some universities, dissertations are required to carry a "declaration" to the effect that the author states to have composed the thesis herself or himself. I find this a nice piece of formality, and therefore, dear reader, please pardon me for declaring the very obvious: the 435 pages of the main text of this thesis have been written by myself, not by anybody else. However, it is *not* plainly a formality when I feel the immediate urge to add: *But* I have enjoyed often invaluable help from my supervisors, and from many others as well. Help from people who have made it possible for me to start with my PhD-project in the first place, who have created or contributed to a productive environment for research, who have advised and encouraged me, or who have helped in another way; all enabling me, in their share, to do the basic research, to write down this part or that, to carry on exploring and writing, and finally, to complete this booklet. Without such help, or even without some parts thereof, my task would have been a very much harder one indeed. It is therefore that I want to acknowledge, below, some of the most important elements of support from which I have benefited over the last nearly five years.

Some parts of this thesis originate from two reports and two papers I have written earlier. So is Chapter 6 an extended version of the paper [Gra02b] and the report [Gra02c]. Both Chapter 4 and Appendix B have grown out of work done in the report [Gra03a], and for the poster [Gra03b] that is summarized by the paper [Gra04b]; however, in each of Chapter 4 and Appendix B also a considerably more involved situation is covered[1]. All other parts of the thesis are "new" in the sense that they have not been published elsewhere.

Acknowledgements

First and foremost my thanks go to my supervisors Jan Willem Klop and Roel de Vrijer: for accepting me, in the first place, as an *AIO* in the Theoretical Computer Science group at the Vrije Universiteit (in early 2000), for helping me to find a place within this group, for their interest in my work, and for their continued support at many levels. (As an important note on the side, I want to mention that I am still

[1] Namely the situation of derivability and admissibility of inference rules in *natural-deduction systems*. Contrasting with this, both the mentioned report and the paper deal with rule derivability and admissibility only in (pure) *Hilbert systems*.

thankful to Dick de Jongh, the head of the Master-of-Logic program of the ILLC at the Universiteit van Amsterdam in 1998-99, when I was part of that program, for suggesting to get in contact with people from the VU.)

I am particularly thankful to Jan Willem for letting me work on his observation of a 'duality' between the Brandt-Henglein and Ariola-Klop systems (which has led, quite directly, to the article [Gra02b], to the report [Gra02c], and to Chapter 6 of this thesis). It happened very often, in late afternoons, when Jan Willem came to our room (Jeroen Ketema's and mine) to take care of his and of my plants, that I could take the opportunity to tell him about this aspect of my work or that. Jan Willem would always listen and give stimulating and, if it was necessary, also critical comments (critical ones, for example, when the language I wrote was too involved, or the titles of my talks and papers were too long). I am very thankful to him for this, and for always indicating possible directions for further investigations. Also, I have the desire to mention that I very much enjoyed assisting Jan Willem in his course on term rewriting systems in two successive years, and once also in a course on process algebra. I think I have learned a lot of things from him in this way; and I want to say my thanks to Jan Willem for supporting me during these first *werkcolleges* that I have taught. Concerning a specific topic in this thesis, I want to acknowledge what is actually not mentioned in the text: that I have been led to Theorem 3.9.14 on page 64, which I hope provides a nice explicit characterization of the "generated subterms" of a recursive type via its "subterms", only by a little example of Jan Willem written down on two tiny sheets of A6-format paper. For all that and for much more: *Thank you very much, Jan Willem!*

Many thanks from me also go to Roel de Vrijer for his interest in my work, and in particular for the regular meetings that I have had with him during my last year as an *AIO*. These meetings (Jan Willem was sometimes present, too), at which Roel let me have total access to his blackboard ("*Alles mag weg!*"), provided me with valuable feedback on what I had written, and they frequently translated into concrete improvements. Also, I want to acknowledge that I was led to the introduction of "Abstract (Pure) Hilbert Systems" and "Abstract Natural-Deduction System", the two topics in Appendix B, by some (as I then felt) quite insistent questions and critical comments by Roel about an abstract setting I had chosen for gathering general results about the notions of rule derivability and admissibility in Hilbert systems. But ultimately also these comments have contributed to make me explore the mentioned topic in a much better framework (in [Gra03a], [Gra03b], [Gra04a] as well as in Chapter 4 and in Appendix B of this thesis). *Thanks a lot for all of that, Roel!*

Also, I want to thank the reading committee for this thesis, Wan Fokkink, Dick de Jongh, Bas Luttik, and Albert Visser, for agreeing to the task of reading through this admittedly voluminous thesis. I am grateful to Wan Fokkink for a long list of corrections, and to Bas Luttik for many comments, some critical remarks, and for many suggestions of improvements (as well as for the readiness to discuss these).

On a very different level, I am indebted to Bas Luttik also in his former function as postdoc in our group: Bas regularly read drafts of my papers and always provided me with extensive and detailed comments. My feeling is that I have learned very much from Bas, on the one hand, during discussions over specific topics in mathematics and computer science, and on the other hand, from many of his suggestions and critical remarks. Bas' comments were frequently connected with the very practical and very important things of how to write down formally technical matters in a stringent, digestible, and non-confusing way. During much of the last year, when revising already existing text for this thesis, and writing other parts for the first time, some of my regular worries were: What would Bas think of this part or that? How would *he* expect a topic in question to be written down *properly*? – Also, I remember fondly the time in the fall semester 2002 when I was able to assist Bas during his course on "Advanced Logic" (*het college "Voortgezette Logica"*) by giving the accompanying practicum; and I am grateful for his support when, at times, I was not satisfied with myself. For all of this, for encouraging me to take the *Staatsexamen NT2*, and for a lot more, let me say: *Thanks very much, Bas!*

I am also very thankful to Jan Rutten for the opportunity he gave me to assist him twice, in successive years, by teaching *de werkcolleges* for his course on "Algebraic Specifications" (in which Jan treats the nice topic of coalgebraic specification of streams and of functions on streams). I have enjoyed it a lot, and I want to thank Jan for his support. I have profited a good deal from discussions with Jan after his lectures, and also from his interest in my work. Furthermore, Subsection 9.2.4 in Chapter 9, concerned with a possible application of a proof-transformation described in Section 8.1 of Chapter 8, has been stimulated by a talk of him in our group's weekly seminar. Also, I want to thank Jan for having had a short look at this subsection and for a couple of valuable comments about it. For all that: *Thank you very much, Jan!*

I am very grateful to Femke van Raamsdonk for letting me organize the weekly seminar of our group twice when she was on leave. Each time this has been a very nice experience for me, and I am thankful for the opportunity of being able to contribute to this regular meeting point of our *sectie Theoretische Informatica* at Friday afternoons. Apart from that, I want to thank Femke for being a tutor for me regarding the writing of referee reports; and much more general, for always being helpful in many respects, for prompt replies to my e-mails, and her regular lively stories about her child(ren). *Thanks very much for that, Femke!*

Thanks are also due to Freek Wiedijk, for causing me a lot of work: he warned me that if the thesis were typeset in 11pt fonts (as I had done so until a short time ago) and if it were then reduced to B5-format by the printers, then the resulting actual font size would be smaller than 9pt and thus really small indeed. This led me to change the font size for my document to 10pt and the format to 170 × 240 mm such that the output did not have to be reduced by the printers. However, it took a big last-minute effort to keep the shape of the layout intact. But nevertheless: *Thank you, Freek!*

My thanks go also to Wan Fokkink, the current head of our group and my roommate during the first half of 2001 (though for one day a week only). In particular, I am indebted to Wan for providing me, much more than once, with extremely helpful information about procedural matters connected with a *promotie*, information that he always illustrated with examples from the cases of his three *promovendi* last year. Also, I thank Wan for helping me out twice recently by agreeing to give a talk in our seminar. On a more general level, I am thankful for his interest in how things are going, and for frequently reminding me of the necessity to submit papers (if there should be any earnest in a scientific ambition—one such reminder has motivated me to start writing [Gra02b]), and for his suggestion to extract perhaps more articles from this thesis (which I hope I'll be able to do). *Thank you very much, Wan!*

And I want to say thanks to my present and former immediate colleagues Mirna Bognar, Jeroen Ketema, Paulien de Wind, and Helle Hansen. My thanks go first to Mirna for providing me with useful information when I was a new member in the group, and for many discussions (even if we sometimes disagreed on this or on that issue). *Many thanks for that, Mirna!*

Jeroen has been my roommate for three years. I have always been impressed by his vast knowledge of many things, of both the theoretical part of computer science and the its very practical aspects (of what *really* happens when one tries to execute this or that command). I have undoubtedly profited from his company over the years and his help in many situations and I am thankful for that (not long ago, Jeroen helped me to install an urgently needed LaTeX-package for typesetting the thesis in the right format). *Thank you very much, Jeroen!*

To Paulien I am very thankful for more things than I can write down here properly. I have frequently searched her advice in matters concerning the Dutch language, in how to write more formal letters or e-mails, and more general, in situations where I was uncertain, because of my foreign background, of how to proceed in a sensible way (but yes, I have also consulted her with mundane practical questions such as ones about subtleties of XEmacs). Paulien always listened very carefully, and then came up, either immediately or a little later, with very thoughtful suggestions for solutions. Furthermore, she taught me how a "supported head-stand" has to be performed correctly (the interested reader could be referred to `http://www.yogajournal.com/poses/481_1.cfm` for the details). And also, I must mention that I was and still am very thankful to her and to Bart for having helped me move to my flat in Diemen last year in an "Ouke Baas" van rented and safely driven by Bart; their help really mattered much to me in that situation and I still value it a lot. *Thank you very much, Paulien!* (*And thank you, Bart!*)

My thanks and greetings go also to my new roommate Helle, for regularly telling me about her work, for explaining me some basics about tableau methods in modal logics, for her good company, and for the fact that I think we do not find it a problem to have to work in a rather small room. Also, more than once I have appreciated very much her energetic and effective help, for example in a situation when a dysfunctional beamer cable nearly rendered most of my preparation for a talk unusable. *Many thanks, Helle!*

Also I want to thank Mihály Petreczky for often dropping by when he was at the VU, for telling me about his work in systems theory, for always being ready to discuss a mathematical topic, and for teaching me about the use of formal power series in the theory of formal languages and automata theory. *Thank you very much, Mihály!*

I am also thankful to Natalia Silvis-Cividjian, for many conversations, for always being interested in how things were going, for providing regular updates on how her son Tigran is doing, for sending me to the course *Academisch Nederlands* in 2003 together with some of the students she was counseling at that time; and perhaps most importantly, for the possibility to exchange thoughts with a *kameraad-allochtoon* about things that often have surprised the both of us concerning the way how things are done in the Netherlands (in contrast to what each one of us had previously been used to in, respectively, Romania and Austria). And lastly, my apologies go to Natalia for having had to come back on a word I had given, cancel an invitation and twice ask for a raincheck for a lunch with her, Peter, Tigran, Paulien and Bart, because of some big last-minute typesetting troubles with this thesis. *Apologies, Natalia, but more importantly: many thanks!*

For some decisively important help while writing the Summary in Dutch of the thesis I want to thank Michel Oey, Jan Willem Klop, and Roel de Vrijer. It was Michel whose persistent argumentation that such a summary should form an independently readable part of a thesis stimulated me to (at least) try to write a text that is as self-contained as possible and that is not merely a short version of the Introduction. Michel pointed me, as so did Roel and Jan Willem later on, to many mistakes, and to sentences and phrases that simply were not Dutch (but had been, most likely, bad translations from German). Eventually, Jan Willem allowed me to continue submitting successive versions for corrections in what was set up to be an approximation procedure aiming at a readable Dutch text. I have to take it as my fault that convergence of this procedure was rather slow and that time did ultimately not suffice to reach a zero of the number-of-corrections function. *Still, I want to express my sincere gratitude for all of the help in this matter!*

Another great debt of mine is to Michel Klein, who was so very kind as to use his expertise with Adobe InDesign for the purpose of turning my rough ideas for the cover of the thesis into a concrete pdf-file. Apart from that, I have repeatedly been able to profit from his experience concerning the printing process of a thesis, and in particular, from his suggestion of Ipskamp as a reliable “print partner”. Finally, Michel gave me, at a moment when that really mattered to me, the LaTeX-commands needed for typesetting the whole document in 170 × 240 mm format. *Thanks very much, Michel!*

It is also my desire to acknowledge the frequent competent help that I have received over the last years by Gert Huisman from the departement’s (and the faculty’s) ICT-group. Whether I had problems with my laptop, or my SUN-workstation did not let me login, or the postscript-file of this thesis had not been accepted by the printers (which was when I found myself without any idea at all about how to carry on from there), I always received very helpful information from Gert which located the problem and made it possible for me to find one or the other way out and forward. I am very thankful for that! My thanks are also due to Ruud Wiggers, who often helped me by directing me to the responsible person, and to many of the young people from the helpdesk who also did their very best in helping me.

Concerning Chapter 4, in which the notions of derivability and admissibility of inference rules are studied, I want to acknowledge the influence that the book [HS86] of Hindley and Seldin had on me when writing the report on which this chapter is based. For quite some time two lemmas in [HS86] provided a constant reminder to me that it must be possible to find some precise statements about connections between the notions of rule derivability and admissibility and my work on proof-transformations. I want to thank Roger Hindley very much for his interest in my work and for an exchange of e-mails in autumn 2003. It is through his mails that I was led to some additional references regarding the history of the notions "derivability" and "admissibility" of rules (that found their place in Remark 4.2.2 on page 81).

I also want to mention some of my colleagues from the *Onderzoeksschool Logica* with whom I got in contact through regular meetings such as the *OzsL-Schoolweek* and the yearly *Accolade*. Please let me just mention a few of them: Nick Bezhanishvili, Annette Bleeker, Rafaella Bernardi, Balder ten Cate, Francien Dechesne, Juan Heguiabehere, Eva Hoogland, Joost Joosten, Clemens Kupke, Fabrice Nauze, Marc Pauly, Yoav Seginer, Merlijn Sevenster, and Willemijn Vermaat. Special thanks go to Francien for inviting me for a talk to Eindhoven, and to Balder and Juan for accepting an invitation for a talk at the VU. My greetings go also to Sebastian Brand for the conversations we regularly had once a year at *NTVI-Theoriedagen* in Utrecht (sometimes remembering our time together at *Cleyndertweg 29* in 1999).

Then I want to express my thanks to many students of *practica* I have given, and especially to those with whom I stayed in contact afterwards; I want to mention Joris, Laurent, Erik, Francis, Ingmar, and quite certainly, Daphne. And as someone, not a student, from quite a distance (not only spatially) apart from the university, I have also promised myself to mention Youssra from de Baarsjes here (whom I came to know in Bos and Lommer).

Furthermore it is my desire to mention the children playing in the courtyard just outside of my flat in Diemen. Their energy, enthusiasm, and (on the whole) their tolerance of each other and care for the younger ones have made it much easier for me to get through Saturday and Sunday afternoons last year with beautiful weather outside, staying inside, busy with typesetting this or xfig-ing that.

Thanks and greetings also go to my landlords in Amsterdam, Aad Wolsink and José Antonio Castro. From Aad I learned a lot about the Netherlands, perhaps most importantly, while learning Dutch I had the big advantage of the daily practice of our conversations (about sport, television, music, cars, and ... all the rest of life). *Thank you very much, Aad!* And to José I am thankful for frequently cooking for me; for helping me to find a good *huisarts*; for being worried about me in situations like when the streets and *de fietspaden* were icy and I was not at home by the usual time; for his frequent translations from and explanations about the *Cerca De Ti* show on Spanish television; and most of all for his undoubtedly *groot en goed hart* that only sometimes was hidden by a shadow. *Hartelijk bedankt, José, voor alles, en mijn beste wensen!*

Towards the end of this long list of thank-you's I want to convey my thanks to my parents Erwin and Herta Grabmayer, to my father in German, and to my mother in English (because she understands and speaks it quite well, and only lacks a bit of practice recently).

Danke, Vater, für vieles, und für viel mehr als ich hier aufzählen kann: Für's Anfertigen-Lassen der beglaubigten Kopien, die ich fuer die Promotions-Formalitäten nötig hatte; für's Schicken der Trainingkleidung, der Handschuhe, und des Pulsfrequenz-Messers; fürs regelmässige Anrufen wenn Mutter in Wien ist; dafür, dass Du mich immer über das sportliche Geschehen in Österreich und das Abschneiden unserer Langläufer auf dem laufenden gehalten hast; und natürlich für die 14 Tage, die Du letztes Jahr zusammen mit Mutter und mir in das Bewohnbar-Machen meiner Wohnung in Diemen investiert hast, mit dem Verlegen von mehr als 60 m^2 *Teppichfliessen und eines Küchenbodens verbracht hast, mit dem "Ausweissigen" der Wände und des Plafonds, mit dem Durchführen von Elektroinstallationen, und mit einer ganzen Reihe von Dingen mehr. Für das alles und mehr: Ein aufrichtiges "Danke Schön"!*

Also, I will not be able to exhaust the long list of things for which I am thankful to my mother, but at least let me give it a try: for a lot of phone-calls that have been much too long (in my father's opinion); for a lot of e-mails from Vienna telling me about her situation and asking me about mine; for regular parcels at Christmas and Easter; for a lot of books that helped me to keep my intellectual balance; recently, for much information about Elfriede Jelinek; for keeping me informed about the situation at Austrian universities, and in Austrian politics; for offering help from the psychological knowledge she acquires during her studies; for letting me stay at her appartement in Vienna during CSL'03; for the idea of having *Fiaker*-trip through the inner city of Vienna together with Bas, Simona, and me; for the mountain tours on Plöckenstein, on Feuerkogel, on the Rax and on Schneeberg; and obviously, also for all of her participation in the dusty work in my flat in Diemen during the two weeks in summer last year! *Danke, Mutter, für alles, und meine besten Wünsche zur Fortsetzung und zum Abschluß Deines eigenen Studiums (auch wenn es noch dauert, ich bin überzeugt, dass Du es schaffen wirst!).*

On a different, also Austria-related note, I still want to thank Prof. Alexander Leitsch from the TU-Wien (my supervisor with my *Diplomarbeit*), and Prof. Konrad Kiener from the JKU-Linz for their help with the application for a scholarship that made it possible for me to come to the Amsterdam in 1998 (it is perhaps not merely a coincidence that both of them know the Netherlands very well).

Last, but not least I want to thank Mihály Petreczky and Paulien de Wind for their readiness to act as my *paranimfen* at the defense of this thesis. I am happy that both of you have agreed to this. *Thanks, P'lien and Mihály!*

Thank yous, well where do I finish?[2] – I have to, now, here.

Clemens Grabmayer Diemen/Amsterdam, January 2005

[2] The question *Thank yous, well where do I start?* is asked by Melanie C in the booklet accompanying her CD "Reason".

Chapter 1

Introduction

Recursive types are prime examples of cyclic objects occurring in computer science. Two binary relations have been studied on classes of recursive types by a sequence of authors (among these are [CaCo91], [AmCa93] and [BrHe98]): an equality relation, frequently called “recursive type equality” or “strong recursive type equivalence”, and a subtyping relation on recursive types. Although algorithms for deciding, whether or not two recursive types are either equivalent or in the subtype relation, have existed for quite some time, formal systems that allow the logical treatment in a formal system of the equality and subtype relations between recursive types are much more recent. Of these systems we have chosen those concerning the equality relation as the objects of our investigation.

1.1 Proof Systems for Recursive Type Equality

Formal axiomatizations that are sound and complete for the equality and subtyping relations on recursive types have been given first by Amadio and Cardelli in [AmCa93]. The authors of that paper also present an algorithm for deciding whether or not two recursive types are in the subtyping relation[1] and establish a connection between the decision algorithm and the formal systems. Their axiomatization for the subtyping relation on recursive types is built on top of an axiom system for recursive type equality. The axiomatization of recursive type equality is hereby of a ‘traditional’ kind[2] with the distinguishing feature of the presence of a *unique-fixed point rule* with applications of the form

$$\frac{\tau_1 = \tau[\tau_1/\alpha] \qquad \tau_2 = \tau[\tau_2/\alpha]}{\tau_1 = \tau_2} \quad \text{(if } \alpha \text{ is “guarded” in } \tau\text{)}$$

[1] Recursive type equality can be decided with the help of this algorithm, but it is better decided by a similar, albeit simpler, algorithm.

[2] In [BrHe98] the axiom system by Amadio and Cardelli is called a “classical axiomatization”.

(where τ, τ_1 and τ_2 are recursive types and α is a type variable) which is related to different forms of *fixed-point rules* that make part of the logical apparatus of a number of similar proof systems: the axiomatizations for the algebra of regular events due to Salomaa in [Sal66], for the notion of 'regular behaviour' in process algebra introduced by Milner in [Mil84], for Kleene algebras and algebras of regular events given by Kozen in [Koz94], and for bisimulation equivalence of cyclic term graphs denoted by μ-expressions presented by Ariola and Klop in [ArKl95, the report version].

Alternative axiomatizations for the equality and subtyping relations on recursive types were later given by Brandt and Henglein in [BrHe98], which "are motivated by coinductive characterizations of type containment and type equality via simulation and bisimulation, respectively". The system for the subtyping relation allows, citing the authors again, "a natural operational interpretation of proofs as coercions" (*coercions* are type-adaptation functions between recursive types that are in the subtyping relation). The main particularity of these proof systems are rules that formalize certain forms of circular reasoning. In a natural-deduction formulation, the system for recursive type equality contains a rule ARROW/FIX that allows applications of the form

$$\dfrac{\begin{array}{c}[\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2]^u \\ \mathcal{D}_1 \\ \tau_1 = \sigma_1\end{array} \qquad \qquad \begin{array}{c}[\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2]^u \\ \mathcal{D}_2 \\ \tau_2 = \sigma_2\end{array}}{\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2}\ \text{ARROW/FIX}, u \tag{1.1}$$

where τ_1, τ_2, σ_1 and σ_2 are recursive types (and where → is the type constructor for function types), and where the assumption classes $[\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2]^u$ at the top of the immediate subdeductions $\mathcal{D}_1$ and $\mathcal{D}_2$ of the deduction in (1.1) are discharged at the displayed application of ARROW/FIX (as indicated by the assumption marker u that is attached to this application). By such a rule application a formula $\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2$ is derived as the conclusion of a deduction $\mathcal{D}$ of the form (1.1) in which $\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2$ may also have been used as an assumption, but where the conclusion of $\mathcal{D}$ eventually does not depend any more on (some or all of the) assumptions of the form of its conclusion. Undoubtedly, the rule ARROW/FIX describes a form of circular reasoning that may appear unsound at first sight, or at least paradoxical. The soundness with respect to recursive type equality of deductions enabled by this rule is ultimately due to the following fact: in an application of ARROW/FIX, assumptions of the form of the conclusion $\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2$ are *not* discharged from an arbitrary deduction with this conclusion, but only from a deduction with conclusion $\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2$, with an application of ARROW/FIX at the bottom, and with immediate subdeductions that respectively end with $\tau_1 = \sigma_1$ and with $\tau_2 = \sigma_2$ (the formulas that result from the conclusion formula by respectively equating the left and the right components of the composite recursive types $\tau_1 \to \tau_2$ and $\sigma_1 \to \sigma_2$ there); deductions of this form have the special property of being 'contractive' according to a definition by Brandt and Henglein.

Apart from the mentioned two kinds of axiomatizations for recursive type equality, we will also consider a 'syntactic-matching' proof system for equational testing that is analogous to one that was introduced by Ariola and Klop in [ArKl95] for the notion of bisimulation equivalence on cyclic term graphs. A specific feature of this system is the presence of a *decomposition rule* (or "deconstruction rule") that allows applications of the form

$$\frac{\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2}{\tau_i = \sigma_i} \quad (\text{for each } i \in \{1,2\})$$

for all recursive types τ_1, τ_2, σ_1, σ_2 and the composite types $\tau_1 \to \tau_2$ and $\sigma_1 \to \sigma_2$. This system does not axiomatize recursive type equality, but it allows to test the "consistency" with respect to strong equivalence of given equations between recursive types. It is sound and complete for recursive type equality in the sense that "contradictions" become derivable if and only if an equation between two recursive types that are *not* strongly equivalent is allowed to be used in derivations as unproven assumption.

1.2 Relating Proof Systems for Recursive Type Equality: Motivations and Aims

In the present study we set out to explore proof-theoretic connections between the three mentioned proof systems for *recursive type equality*. Our initial motivation consisted in an observation of J.W. Klop, who for slightly more general proof systems recognized the following: a striking similarity between the activities of (a) trying to demonstrate the consistency of an equation relative to a certain syntactic-matching system by a 'loop-checking' procedure, and of (b) trying to derive the same equation in an appropriate Brandt-Henglein-like proof system. Since this observation could easily be reformulated in relation to the respective proof systems for recursive type equality, the question arose of whether the purported close relationship of problems could indeed be formulated as a precise statement, and as such be proved, at least in the somewhat simpler situation considered here of proof systems for recursive types.

By this apparently very close relation between the syntactic-matching and Brandt-Henglein systems also the question was initiated whether there did exist also other interesting proof-theoretic relationships between the mentioned proof systems for recursive type equality. A particularly intriguing goal was to find out more about connections between the coinductively motivated system of Brandt and Henglein and the system of Amadio and Cardelli. Or, put differently, to find precise answers to the question of whether, and if so then how, coinductive reasoning that is formalized by derivations in the Brandt-Henglein system could be translated into the more 'traditional' form of reasoning as formalized by derivations in the system of Amadio and Cardelli; and vice versa, whether, and if indeed then how, a translation in the opposite direction could be achieved.

The aim we pursue here is to contribute some concrete answers to questions of the kind just mentioned. The main theorems that we will give state *connections by effective proof-theoretic transformations* between the three kinds of proof systems mentioned in Section 1.1 (and a number of additional variant systems); these transformations will be established in the proofs as stepwise and effective operations on formal derivations.

The predominant part of transformations that we will give are part of what [TS00] call *interpretational proof theory* and describe as having “syntactical translations of one formal theory into another” as its tools. Only one operation on proofs that we shall work out may be viewed as being part of *structural proof theory* that, as [TS00] formulate, is concerned with “a combinatorial analysis of the structure of formal proofs” and counts cut-elimination and normalization to its central methods.

1.3 Overview

Below we give a brief overview of the respective contents of the eight chapters following this introduction, and of the three appendices. For precise page references regarding respective chapters, appendices and sections we want to direct the reader to the table of contents on page i.

1.3.1 The Chapters

In **Chapter 2** we gather frequently used mathematical notation; and we explain aspects of the notation used in our formal treatment in later chapters of derivations, i.e. formal proofs.

In **Chapter 3** we set out to give a rather detailed survey of notions and notations that are related to the formal treatment of recursive types. As the objects on which our study is based, we formally introduce a restricted class of recursive types denoted by μ-expressions with $\to$ as the only type constructor. We also introduce the subclass of all recursive types that belong to the mentioned class and that are “in canonical form”. It will be necessary to adopt suitable terminology and conventions for the treatment of substitution expressions and of the variant relation (renaming of bound variables) on recursive types.

Then the definition of the central notion of the “tree unfolding” of a recursive type is given, which is able to provide ‘unwinding semantics’ for recursive types: in it two recursive types are considered to be equal if and only if their tree unfoldings are the same. This semantics gives rise to the equivalence relation “strong recursive type equivalence”, or “recursive type equality”, on recursive types. Another relation, “weak recursive type equivalence”, is defined as the smallest congruence relation that is generated by the basic operations of “folding” and “unfolding” on recursive types; it will be shown that this relation is in fact weaker than in strong recursive type equivalence. Furthermore, an effective and natural transformation of recursive types into recursive types in canonical form is given. And eventually the notion of "generated subterm" of a recursive type is introduced, and later needed facts about this concept are collected.

In **Chapter 4** we gather the definitions and a couple of basic facts about the notions of derivability and admissibility of inference rules in two kinds of formal systems that we will encounter later on: in "pure" Hilbert systems, and in natural-deduction systems. We explain the definition of rule derivability and admissibility in these systems, and collect some useful basic facts about these notions. And in particular, we give results concerning the relationships between the notions of rule derivability and admissibility (in the respective kind of systems) and the possibility to eliminate rule applications from derivations. Statements concerning such relationships will be of some use in later chapters, if only as background knowledge that is able to help our understanding in certain situations.

In **Chapter 5** we formally introduce, on the one hand, the axiom systems for recursive type equality due to Amadio and Cardelli and due to Brandt and Henglein, and on the other hand, a 'syntactic-matching' system that is fit for equational testing with respect to recursive type equality; this latter system is an adaptation of a similar 'syntactic-matching' system that was given by Ariola and Klop in [ArKl95]. We report (and partly also sketch proofs for) the soundness and completeness theorems of these systems with respect to recursive type equality. Furthermore, we define respective 'analytic', or 'normalized', variants of the system by Brandt and Henglein, and of the syntactic-matching system. Although being closely related to the original systems, the variant systems possess stronger proof-theoretical properties as a consequence of the absence in them of the transitivity rule; in particular, the variant systems enjoy respective "subformula properties". Still in Chapter 5, we gather basic facts about the difference in proof-theoretical properties between the axiom systems on the one hand and the syntactic-matching systems for equational testing on the other hand. For instance, we examine the 'theory' of the syntactic-matching systems, and we ask what notions of relative consistency are induced by each of the axiom systems.

In **Chapter 6** we investigate the mentioned observation by J.W. Klop regarding a similarity between the problems of checking the consistency of a given equation with respect to the syntactic-matching system, and of finding a proof in the axiom system of Brandt and Henglein. As a start, we show that this similarity can be described particularly well for the earlier defined 'analytic' variants of the Brandt-Henglein and syntactic-matching systems (in fact, the search for an easy way to put this observation has provided a strong motivation for defining these variant systems). For extracting a precise statement, however, a couple of further prerequisites will be needed. Firstly, we introduce a conservative extension of the variant Brandt-Henglein system by adding some more rules of a circular nature. And secondly, we define "consistency-unfoldings" in the analytic syntactic-matching system as certain downwards-growing derivation-trees that formalize successful consistency-checks with respect to this system. Relying on these notions, our main result in Chapter 6 will then consist in the following assertion: there exists a *duality* between derivations in the extended variant Brandt-Henglein system and consistency-unfoldings in the variant syntactic-matching system via easily definable reflection mappings. And what is more, this duality between derivations and consistency-unfoldings can *geometrically* be *visualized*.

The following two chapters, Chapter 7 and Chapter 8, are devoted to the goal of developing effective transformations between derivations in the axiom systems of Amadio and Cardelli and of Brandt and Henglein. Neither of the transformations that we will describe are of an entirely straightforward nature, for which there are reasons, likely to be inherent ones, connected with essential differences in the logical features of these systems. The most important distinction consists hereby in the fact that the Amadio-Cardelli system is a pure Hilbert system, whereas our formulation of the Brandt-Henglein system is a natural-deduction system (in [BrHe98] it has been introduced as a sequent-style Gentzen-system).

In **Chapter 7** we build a transformation from derivations in the Amadio-Cardelli system into respective derivations in the Brandt-Henglein system. In the first of the two sections of this chapter we carry out some preparations: we obtain further basic facts (beyond those already treated in Chapter 5) about the 'proof theory' of the Amadio-Cardelli system. We show that three kinds of substitution rules are admissible in this system, and that applications of these rules can always be removed, by an effective procedure, from derivations in the extension of the Amadio-Cardelli system with the substitution rules. And then we prove that a particular rule of the Amadio-Cardelli system, the μ-compatibility rule, can be dispensed with in a close variant system, or to be more precise, it is an admissible rule in a variant Amadio-Cardelli system; we also explain how to eliminate applications of this rule effectively. And subsequently in the second and last section of this chapter, we develop an effective translation of such derivations in the Amadio-Cardelli system that do not contain applications of the μ-compatibility rule into derivations in the Brandt-Henglein system. Together with results obtained in the first section this will eventually establish the existence of an effective transformation of derivations in the Amadio-Cardelli system into derivations in the Brandt-Henglein system.

In **Chapter 8** we develop an effective transformation in the opposite direction, namely, from derivations in the Brandt-Henglein system into derivations in the Amadio-Cardelli system. Such a transformation will be built in two steps. Firstly, we give an effective transformation from derivations in the analytic variant Brandt-Henglein system via derivations in an annotated version of this system into derivations in the system of Amadio and Cardelli. And secondly, we complement this transformation by one that is able to 'normalize' an arbitrary derivation in the Brandt-Henglein system with the outcome of a respective derivation in the analytic variant of this system.

In **Chapter 9**, the conclusion, we summarize the transformations developed in previous chapters, and then try to get them into a somewhat broader perspective. We give a figure that shows how the proof systems belonging to one of the three considered groups are linked by the main ones of our transformations, and we discuss some noticeable features of our transformations in the light of this picture. And eventually, we describe four directions for possible extensions of our results.

1.3.2 The Appendices and Indices

In **Appendix A** we gather some of the more technical proofs for statements in Chapter 3. There, proofs are contained from the sections regarding substitution expressions involving recursive types, the variant relation on recursive types, the tree unfolding and the leading symbol of a recursive type, and the generated subterms of a recursive type.

In **Appendix B** we gather precise versions of statements reviewed in Chapter 4 about the notions of derivability and admissibility of inference rules. We introduce the notions of "abstract pure Hilbert system" (APHS) and "abstract natural-deduction system" (ANDS), adapt the notions of rule derivability and admissibility to this systems, and collect basic facts about them. Hereby we are in particular interested in results concerning the relationship between rule derivability and admissibility and the possibility to eliminate applications of rules from given derivations.

In **Appendix C** results are stated and proved that assert a bound on the depth of such derivations in the analytic version of the Brandt-Henglein system that do not contain certain kinds of redundancies. These results are designed for their use in the proof of statements in Chapter 7 and in Chapter 8; in particular, they are invoked for showing termination of procedures that build up "redundancy-free" derivations in the variant Brandt-Henglein system in a bottom-up manner.

The **Bibliography** is given starting on page 415. On page 419 it is then succeeded by the **Index of Notations**, which contains page references to the elements of notation we use; a **Subject Index** follows, beginning at page 423. Finally, a **Summary in Dutch** (the **Samenvatting**) is given on page 429.

Chapter 2

Preliminaries

In this chapter we gather some of the more frequently used basic notions and notations. In Section 2.1 we explain how we designate and treat basic mathematical objects and entities, and in Section 2.2 we gather the basic proof-theoretic terminology and notation that we will use.

2.1 Basic Mathematical Notions and Notation

Some general abbreviations that will be used here are "iff" for "if and only if", "i.e." for "id est" (latin for "this is" or "this means"), and "w.l.o.g." for "without loss of generality".

We will use the symbol $\equiv$ for *literal identity* of formal expressions (such as strings over a given alphabet, see Subsection 2.1.4, or recursive types, see Definition 3.1.1, Section 3.1, Chapter 2). The symbols $=_{\mathrm{def}}$ and $\Leftrightarrow_{\mathrm{def}}$ are used to indicate definitions: for example, in an expression of the form $s =_{\mathrm{def}} t$, the object s is defined as the object t, and in a stipulation of the form $P \Leftrightarrow_{\mathrm{def}} Q$ the property P is defined as the property Q. In definitions of formal grammars we will however use the symbol $::=$ for defining the non-terminal symbol on the left-hand side by a sequence of construction clauses, separated by the symbol $|$, on the left-hand side.

We use the symbol □ for marking the end of proofs; and we use ⊠ and ■ at the end of definitions and examples, respectively.

2.1.1 Basic Set-Theoretic Notation

We abbreviate the set $\{0, 1, 2, 3, \ldots\}$ of *natural numbers including zero* by ω, and the set $\{1, 2, 3, \ldots\}$ of *natural numbers* by $\omega\backslash\{0\}$. Standard set-theoretic notations such as, for example, the *membership relation* $\in$, and the *subset relation* $\subseteq$ are used in their usual meaning; the *proper subset relation* is abbreviated by $\subsetneqq$.

Let A be a set. We designate by $\mathcal{P}(A)$ the *powerset of* A, i.e. the set of all subsets of A; by $\mathcal{P}_{\mathrm{f}}(A)$ we denote the *set of all finite subsets of* A. We denote by

$|A|$ the *cardinality* of A (however, only finite cardinalities will occur, i.e. we will encounter expressions $|B|$ only for finite sets B, thereby denoting the number of elements of B).

The *Cartesian product* $\{\langle a,b\rangle \mid a \in A,\ b \in B\}$ of two sets A and B is denoted by $A \times B$; and in particular, pairs are denoted in angle-brackets notation: a *pair* over A and B is a member $\langle a,b\rangle$ of $A \times B$. Angle brackets are also used for *triples* $\langle a_1,a_2,a_3\rangle$, *quadrupels* $\langle a_1,a_2,a_3,a_4\rangle$, ... of elements $a_1,a_2,a_3,a_4,\ldots$ of a set A (so as to ensure that these can be viewed as sequences over A as defined below).

2.1.2 Functions and Partial Functions

The terms *function* and *mapping* are used synonymously. Given that $f : A \to B$ is a function between two sets A and B, we denote by $f(C) =_{\text{def}} \{f(x) \mid x \in C\}$, for all sets $C \subseteq A$, the *image of* C *under* f; and for all $b \in B$, we denote by $f^{-1}(b)$ the set $\{a \in A \mid f(a) = b\}$. The composition $g \circ f : A \to C$ of two functions $f : A \to B$ and $g : B \to C$ is defined, for all $a \in A$, by $g \circ f(a) =_{\text{def}} g(f(a))$.

A *partial function* f between two sets A and B will be denoted by

$$f : A \rightharpoonup B \tag{2.1}$$

(underlying such a partial function is a function on a subset of A); for all $a \in A$, we denote by $f(a)\!\downarrow$, and by $f(a)\!\uparrow$, the statements that f is defined for a, and that f is not defined for a, respectively; clearly, if $f(a)\!\uparrow$ holds, then we denote by $f(a)$ the element of B that is the result of applying f to a. For a partial function of the form (2.1), we respectively denote and define by

$$\begin{aligned} \mathrm{dom}(f) &=_{\text{def}} \{x \in A \mid f(x)\!\downarrow\}\ , \\ \mathrm{ran}(f) &=_{\text{def}} \big\{y \in B \,\big|\, (\exists\, x \in A)\,\big[\, f(x)\!\downarrow\ \ \&\ y = f(x)\,\big]\big\} \end{aligned}$$

the *domain*, and the *range of* f (the symbol "&" occurring in the definition of $\mathrm{ran}(f)$ expresses the logical conjunction of the statements to its left and to its right; such use of "&" for denoting informal conjunctions will be stipulated in Subsection 2.1.6).

2.1.3 Finite Multisets and Sequences

Let A again be a set. By $\mathcal{M}_{\mathrm{f}}(A)$ and $\mathit{Seqcs}_{\mathrm{f}}(A)$ we respectively denote the *set of all finite multisets over* A and the *set of all finite sequences over* A, i.e. we let

$$\begin{aligned} \mathcal{M}_{\mathrm{f}}(A) &=_{\text{def}} \{M : A \to \omega \mid M(a) \neq 0 \text{ holds only for finitely many } a \in A\}\ , \\ \mathit{Seqcs}_{\mathrm{f}}(A) &=_{\text{def}} \{\langle\rangle\} \cup \{\langle a_1,\ldots,a_n\rangle \mid n \in \omega\backslash\{0\},\ a_1,\ldots,a_n \in A\}\ , \end{aligned}$$

where $\langle\rangle$ denotes the sequence of length 0, and where $\langle a_1,\ldots,a_n\rangle$ denotes a sequence of n elements, starting with $a_1,a_2,a_3,\ldots$ and ending with a_n. The union of two finite multisets $M_1,M_2 \in \mathcal{M}_{\mathrm{f}}(A)$ over A is defined and designated by

$$\begin{aligned} M_1 \uplus M_2 :\ & A \to \omega \\ & x \mapsto (M_1 \uplus M_2)(x) =_{\text{def}} M_1(x) + M_2(x)\ . \end{aligned}$$

By $\mathrm{lg} : \mathit{Seqcs}_{\mathrm{f}}(A) \to \omega$ we designate the function that to every sequence $\sigma \in$ $\in \mathit{Seqcs}_{\mathrm{f}}(A)$ assigns its *length* $\mathrm{lg}(\sigma)$ (for example, $\mathrm{lg}(\langle x_1, x_2, x_3, x_4\rangle) = 4$ and $\mathrm{lg}(\langle\rangle) = 0$). Furthermore we define, for all sets A and for all $i \in \omega\backslash\{0\}$, the partial function

$$\begin{aligned} \mathrm{proj}_i :\ & \mathit{Seqcs}_{\mathrm{f}}(A) \rightharpoonup A \\ & \xi \mapsto \begin{cases} x_i & \ldots\ \xi = \langle x_1, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_n\rangle \text{ for some } n \in \omega,\ n \geq i \\ \uparrow & \ldots \text{ else} \end{cases} \end{aligned}$$

on $\mathit{Seqcs}_{\mathrm{f}}(A)$, and we call proj_i the *i-th projection function* on $\mathit{Seqcs}_{\mathrm{f}}(A)$.

Eventually, we define the operations $\mathrm{set}(\cdot)$ and $\mathrm{mset}(\cdot)$ that 'convert' finite multisets or sequences to finite sets, and finite sets or sequences to finite multsets, respectively. More precisely, we define, for arbitrary sets A, the functions

$$\mathrm{set} : \mathcal{M}_{\mathrm{f}}(A) \cup \mathit{Seqcs}_{\mathrm{f}}(A) \to \mathcal{P}_{\mathrm{f}}(A)\ , \qquad \mathrm{mset} : \mathcal{P}_{\mathrm{f}}(A) \cup \mathit{Seqcs}_{\mathrm{f}}(A) \to \mathcal{M}_{\mathrm{f}}(A)$$

in the following way: the function $\mathrm{set}(\cdot)$ assigns to every finite multiset $M \in \mathcal{M}_{\mathrm{f}}(A)$ the finite *set* $\mathrm{set}(M)$ of all elements of A that occur in M (once or more often)[1], and to every finite sequence $\sigma \in \mathit{Seqcs}_{\mathrm{f}}(A)$ the finite *set* $\mathrm{set}(\sigma)$ of all elements of A that occur in σ. And the function $\mathrm{mset}(\cdot)$ assigns to every finite subset Y of A the finite *multiset* $\mathrm{mset}(A)$ in which every element of Y occurs precisely once and no other elements of A occur, and to every sequence $\sigma \in \mathit{Seqcs}_{\mathrm{f}}(A)$ the finite *multiset* $\mathrm{mset}(\sigma)$ in which every element of A occurs precisely as often as in σ and no other elements of A occur.

2.1.4 Strings

Let A be a set. By A^* we designate the set of all *strings* (or *words*) *over alphabet* A including the empty string ϵ. For a string $w \in A^*$, we denote by $|w|$ the length of the string w. Strings of length 1 are identified with the element they contain. For the *concatenation* of two strings u and v over A we write $u.v$; we will however frequently drop the concatenation symbol . from a notation like $u.v$ and write uv instead. For all words $w \in A^*$, we denote by

$$\mathit{Pref}(w) =_{\mathrm{def}} \big\{u \in A^* \bigm| (\exists\, v \in A^*)\, [uv = w]\big\}$$

the set of all *prefixes* of w. Using the symbol $\equiv$, the *literal identity* of two strings $u, v \in A^*$ will be designated by $u \equiv v$.

2.1.5 Reduction Relations

By a *reduction relation* on a set A we understand just a binary relation $\to\ \subseteq A \times A$. An *abstract reduction system* is a structure $\mathcal{A} = \langle A, \to\rangle$ consisting of a set A and a reduction relation $\to$ on A.

[1] For all $M \in \mathcal{M}_{\mathrm{f}}(A)$, $a \in A$ and $n \in \omega$, we say that *a occurs n times in M* if and only if $M(a) = n$.

Let A be a set and let $\to$ be a reduction relation on A. The *inverse relation* of $\to$ is denoted by $\leftarrow$. By $\twoheadrightarrow$ we denote the reduction relation on A that is the *reflexive and transitive closure* of $\to$, which is also called the *more-step reduction relation* with respect to $\to$; by $\twoheadleftarrow$ we mean the inverse relation of $\twoheadrightarrow$. By $\to^+$ we denote the *transitive closure* of $\to$, and by $\leftarrow^+$ the inverse relation of $\to^+$. By $\longleftrightarrow$ we denote the symmetric closure of $\to$, i.e. we let $\longleftrightarrow =_{\text{def}} \to \cup \leftarrow$. And finally, by $\twoheadleftarrow\!\!\twoheadrightarrow$ we denote the *convertibility relation* (also called the *conversion*) belonging to $\to$, i.e. the reflexive, symmetric and transitive closure of $\to$ (in other words, the equivalence relation generated by $\to$).

Analogous designations are used for reduction relations that are denoted by the symbol $\to$ decorated with some name label (for example, $\twoheadrightarrow_{\text{ren/out-unf}}$ and $\twoheadleftarrow\!\!\twoheadrightarrow_{\text{ren/out-unf}}$ respectively denote the more-step reduction relation and the convertibility relation belonging to the reduction relation $\to_{\text{ren/out-unf}}$ that will be introduced in Definition 5.3.5 on p. 138).

2.1.6 Logical Symbols

In definitions we frequently use the standard language of predicate logic with the logical operators $\vee$, &, $\to$, $\leftrightarrow$ (we follow [Shoe67] in using & in place of $\wedge$) and the quantifier $\exists$ and $\forall$. Informal quantifications in definitions, proofs, etc. will be represented similar as in the example $(\forall a \in A)\,(\exists\, b \in B)\,[R(a,b)]$. When referring to proof systems in [TS00] concerned with predicate logic, we will however conform to the precise way how the formula language of predicate logic is defined according to [TS00, p.2] (and then, in particular, use the operator $\wedge$ in place of &).

2.1.7 Trees

In this subsection, we follow the stipulations for trees in [TS00, 1.1.8, p.9]. A *tree* is a partially ordered set $\langle X, \leq \rangle$ with a smallest element and with the property that, for all $x \in X$, the set $\{ y \mid y \leq x \}$ is totally ordered (i.e. it is a linear order). The elements of X are called the *nodes* of the tree; *branches* are maximal totally ordered subsets of X (i.e. subsets that cannot be extended without introducing incomparable elements).

Trees are usually expected to grow upwards, in a nature-like manner; however the derivation-trees introduced in Chapter 6 will be an exception to this rule. If a branch of a tree is finite, it ends in a *leaf* or *top node* of the tree. If n, m are nodes of a tree with partial ordering $\leq$, and $n < m$ holds (i.e. $n \leq m$ and $n \neq m$ hold), then m is called a *successor* of n, and n a *predecessor* of m. If $n < m$ holds and if there are no nodes in between n and m, then n is called the *immediate predecessor* of m, and m an *immediate successor* of n.

Also labeled trees will be considered (in particular, “prooftrees” will be labeled trees), with functions assigning objects (usually formulas) to the nodes. The terminology for trees also applies for labeled trees.

2.2 Proof-Theoretical Notation

Apart from a few minor exceptions (that will be pointed out clearly), we will base ourselves on the introduction of the basic proof-theoretic notions in [TS00] and on the particular notation for formal derivations developed and used there. The basic stipulations from [TS00] for the formal treatment of such systems are reported and slightly adapted in the sequel below.

Derivations (in [TS00] also called "deductions") are viewed as labeled trees that are presented in a nature-like manner with the root at the bottom; each node carries a formula as its label. The formulas at the immediate successors of a node ν are the premises of a rule application a, the label of ν is the conclusion of a. At the root of the tree we find the conclusion of the whole deduction.

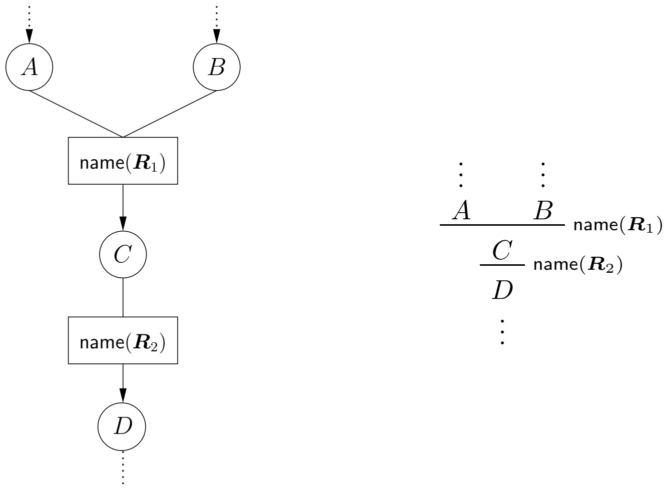
The word *proof* is restricted in its use to arguments on the meta-level; for formal demonstrations either of the terms *derivation* or *deduction* is used. But *prooftree* will mean the same as *deduction tree* and will be used synonymously for "derivation" and "deduction" when we want to emphasize the graphical aspects of the representation of a derivation. We will speak of a *symbolic prooftree* if parts of a prooftree are abbreviated symbolically in ways described in more detail below. We will not use the term *derivation tree* in the way [TS00] do, namely, in the same meaning as "prooftree" and "deduction tree", but we will refer by this expression to certain downwards-growing and downwards-branching labeled "trees of consequences" that we will use in Chapter 6 and that will be defined there in a very similar way as derivations. We will use the symbol $\mathcal{D}$ (possibly sub- or superscripted or with a modification like $\tilde{}$ attached on its top) as a syntactical variable for derivations and the symbol $\mathcal{C}$ for derivation trees.

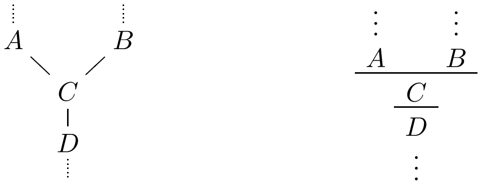
Although this is also in slight contradiction with the suggestion in [TS00] to use the word 'proof' only for meta-level objects, we follow quite common usage of the term *proof system* for formal systems. In particular, we will subsume under it (as we have already done so in the title of this thesis and in the Introduction) both axiom systems and formal systems that are fit for testing the 'consistency' of given formulas.

Rules are generally considered to be schemes consisting of *applications*, *inferences* or *instances* of the rule. However, a different, and more abstract, notion of rule is suggested in Appendix B, where the concepts "abstract pure Hilbert system" (APHS) and "abstract natural-deduction system" (ANDS) are introduced. In the case of APHS's, a rule is a set of instances endowed with a premise and a conclusion function; instances of such rules can be viewed as "hyperedges" of the concept of "hypergraph" (for example, see [Plu93]). In the case of ANDS's the situation is slightly more complex.

Derivations are displayed in a way that is slightly different from how labeled trees are usually drawn: if in a derivation the immediate neighborhood of a node with two predecessors and one successor looks like the tree on the left, then this part of the deduction is instead represented more compactly as on the right:

Figure 2.1: Example for the formalization of the common practice of denoting the names of applied rules in derivations by rule name labels (see the indicated derivation on the right-hand side) by viewing derivations as labeled 'hypertrees' (on the left-hand side).





The general practice, however, is perhaps more adequately represented by the way how derivations are introduced in the abstract concepts of proof systems in Appendix B: inferences within derivations are supposed to carry also the names name($\boldsymbol{R}$) of applied (named) rules $\boldsymbol{R}$ (and in the case of derivations in abstract natural-deduction systems, to carry also markers representing classes of open assumptions that get discharged). For instance, in APHS's, where rule applications can be viewed as hypergraph hyperedges, derivations can be represented as hypertrees, i.e. hypergraphs with an 'underlying' tree such that the nodes of the hypergraph are labeled by formulas, and connected by hyperedges that correspond to rule applications and are labeled by the respective rule name. We do not make these notions precise here, but instead refer to Figure 2.1 for an appealing example.

Characteristic for systems of *natural deduction* is the use of assumptions that may be *discharged* (or *closed*) at some later step in the deduction. Assumptions are provided with markers. We will always assume a countably infinite set Mk of *assumption markers* to be given. Markers from this set will be attached to unproven assumptions at the top of a prooftree and will enable to do bookkeeping as to (A) whether or not a particular assumption at the top of a considered prooftree has already been discharged and as to (B) by which rule application in a derivation an assumption, on which the conclusion does not depend any more, has actually been discharged. We will generally use variables u, v, w as syntactical variables that vary through markers in Mk; however, an exception will be Section 8.1 of Chapter 8, where we will define a proof systems in which type variables (indicated by the syntactical variables using small Greek letters $\alpha, \beta, \ldots$) are used as assumption markers.

The notations

$$\begin{array}{c} [A]^u \\ \mathcal{D} \\ B \end{array} \qquad\qquad \begin{array}{c} (A) \\ \mathcal{D} \\ B \end{array} \qquad\qquad \begin{array}{c} \mathcal{D}' \\ [A] \\ \mathcal{D} \\ B \end{array} \qquad\qquad \begin{array}{c} \mathcal{D}' \\ (A) \\ \mathcal{D} \\ B \end{array}$$

—to which we will refer to from left to right by the numbers (1)–(4) —have the following meaning (our understanding of these notions deviates slightly from the explanation given for them in [TS00] on p. 21): (1) denotes a derivation $\mathcal{D}$ with conclusion B and a set $[A]^u$ of open assumptions that consists of all *undischarged* or *open* occurrences of the formula A at the top nodes of the prooftree $\mathcal{D}$ with marker u (*note:* both B and the formulas in $[A]^u$ are part of $\mathcal{D}$ and the set $[A]^u$ may be empty); (2) stands for a derivation $\mathcal{D}$ with conclusion B, in which a particular occurrence of the formula A at the top of $\mathcal{D}$ is singled out by the context (of an argument, a proof, a statement, etc.) in which this derivation occurs (*note:* this notation entails that there does exist at least one occurrence of a formula A at the top of the prooftree $\mathcal{D}$); (3) means a derivation with conclusion B that arises from a derivation $\mathcal{D}'$ with conclusion A and a derivation $\mathcal{D}$ of the form (1) (with some $u \in Mk$) by extending $\mathcal{D}$ above each of the marked assumptions A^u belonging to the assumption class $[A]^u$ by copies of $\mathcal{D}'$ (*note* that the marker u for the assumptions A^u above which $\mathcal{D}$ is extended is not relevant any more); (4) denotes the result of placing a copy of the derivation $\mathcal{D}'$ with conclusion A above the particular occurrence singled out in a derivation $\mathcal{D}$ of the form (2) with conclusion B.

We mention a later frequently used way to shorten complicated prooftrees that is widely used in proof-theory and that dates back to the seminal work [Gen35] of Gentzen: a sequence of two or more, rather straightforward, one-premise applications of rules in which sequence the order of the applications is either obvious or not relevant is allowed to be indicated by a double line. For example, a symbolic prooftree of the form

$$\begin{array}{c} A \\ \hline\hline B \end{array}\; R_1, R_2, \ldots, R_n$$

is allowed to be used for a deduction of the formula B from the formula A by n one-premise applications of the rules R_1, R_2, $\ldots$, R_n, respectively; the order in which these rule applications are carried out is either arbitrary, or it is assumed to be obvious from the rules R_1, $\ldots$, R_n and the context in which such a deduction appears (and usually it is determined by the succession from left to right in the list $R_1, R_2, \ldots, R_n$ of rules attached to such a 'multi-step inference').

In Chapter 4, Section 4.3.3, and in Appendix B, Section B.2, we introduce the notion of "derivation context" in natural-deduction systems: derivation contexts are the respective result of replacing some subderivations of a derivation in such a system by context-holes. By a k-ary derivation context a derivation context with holes among $[\,]_1, \ldots, [\,]_k$ is meant. For a k-ary derivation context $\mathcal{DC}$ and derivations $\mathcal{D}_1, \ldots, \mathcal{D}_k$ with respective conclusions $A_1, \ldots, A_k$, the result $\mathcal{DC}[\mathcal{D}_1, \ldots, \mathcal{D}_k]$ of hole-filling will be visualized as the symbolic prooftree

$$\begin{array}{ccc} \mathcal{D}_1 & & \mathcal{D}_k \\ {[A_1]_1} & & {[A_k]_k} \\ & \mathcal{DC} & \\ & B & \end{array}$$

which is defined by extending the labeled tree corresponding to $\mathcal{DC}$ above each of the occurrences of a context hole $[\,]_i$, where $i \in \{1, \ldots, k\}$, by the derivation $\mathcal{D}_i$.

More specific proof-theoretic notations will be introduced at a number of places later on. Most notably, notation for describing (and handling of) finite downwards-growing derivation-trees is agreed in Notation 6.3.3 and in Notation 6.3.7, Chapter 6. And useful notation for describing the open assumption classes of derivations in natural-deduction systems is stipulated in Notation 8.1.3, Chapter 8.

Chapter 3

Recursive Types

In this chapter we introduce basic definitions regarding the particular, restricted class of recursive types for which we will consider, in a later chapter, proof systems that formalize a notion of equality between recursive types. A first such proof system is already encountered here, but most of them will be introduced only in Chapter 5.

In Section 3.1, we define a restricted class of recursive types in which only $\rightarrow$ appears as type formation symbol. We give definitions for "recursive types", and for "recursive types in canonical form", as terms with a μ-binding as well as for the notions of the syntactical depth and the size of recursive types. In Section 3.2 we introduce the notions of contexts and subterms of a recursive type, together with the technical concept of positions in a recursive type. In the following section, Section 3.3, we assemble definitions and conventions for the way how we will treat substitution expressions involving recursive types. In Section 3.4, we gather definitions, properties and notation for the variant relation between recursive types which is the counterpart of α-conversion on λ-terms for the recursive types in μ-term notation considered here.

The important notions of the tree unfolding, and of the leading symbol of a recursive type are then defined in Section 3.5. Relying on the concept of tree unfolding, in Section 3.6 the notion of "strong recursive type equivalence", also called "recursive type equality" is defined, which is fundamental for proof systems that will be introduced later in Chapter 5. The weaker notion of "weak recursive type equivalence" is subsequently defined and studied in Section 3.7; it is in the definition of this notion that a proof system for a notion of equality between recursive types will be encountered for the first time here. In Section 3.8 we explain a well-known transformation that takes general recursive types to recursive types in canonical form. Finally, in Section 3.9 we define the important notion of generated subterm of a recursive type and gather the most important results related to this notion, which we will need for our proof-theoretical investigations in later chapters.

3.1 Definition

We start with the formal definition of recursive types and recursive types in, so called, canonical form.

Definition 3.1.1 (Recursive types and recursive types in canonical form). Let *TVar* be a countably infinite set of *type variables*

(i) The set μTp of *recursive types* is generated by the following grammar that is given in Backus-Naur-Form:

$$\left.\begin{array}{rcl} \alpha & ::= & \underline{\alpha}_1 \,|\, \underline{\alpha}_2 \,|\, \underline{\alpha}_3 \,|\, \ldots \\ \tau & ::= & \bot \,|\, \top \,|\, \alpha \,|\, (\tau \to \tau) \,|\, (\mu\alpha.\, \tau) \end{array}\right\} \tag{3.1}$$

where $TVar = \{\underline{\alpha}_1, \underline{\alpha}_2, \underline{\alpha}_3, \ldots\}$. $\bot$ is called the *bottom type*, and $\top$ the *top type*. The members of μTp are called *recursive types*. Recursive types that are of the form $\tau_1 \to \tau_2$, for some $\tau_1, \tau_2 \in \mu Tp$, i.e. recursive types which are formed according to the fourth disjunctive clause for the non-terminal symbol τ in the grammar (3.1), are called *composite*.

In a recursive type $(\mu\alpha.\, \tau)$ the μ-operator acts as a binding for all occurrences of the variable α in τ that are not located within subexpressions of τ of the form $(\mu\alpha.\, \tau_0)$ (and which consequently are already bound inside τ by another μ-binding of α). Thus an occurrence of the type variable β in a recursive type σ is called a *bound occurrence* of β in σ iff it is contained in some subexpression $(\mu\beta.\, \rho)$ of σ (i.e. iff for some subexpression $(\mu\beta.\, \rho)$ of σ it either falls within ρ or is just the occurrence of β immediately following the leading μ-operator in $(\mu\beta.\, \rho)$). Conversely, a variable occurrence of a type variable γ in a recursive type σ is called *free* iff it is not a bound occurrence of γ in σ. For every recursive type χ, the *set of bound variables* of (i.e. the set of variables occurring bound in) χ is designated by $\mathrm{bv}(\chi)$, and the *set of free variables* of (i.e. the set of variables occurring free in) χ by $\mathrm{fv}(\chi)$.

(ii) The *set can-μTp of recursive types in canonical form*, a subset of the set μTp of all recursive types, is generated by the following grammar that is given in an informal Backus-Naur-Form:

$$\tau ::= \bot \,|\, \top \,|\, \alpha \,|\, (\tau_1 \to \tau_2) \,|\, (\mu\alpha.\, \underbrace{(\tau_1 \to \tau_2)}_{\text{where } \alpha \in \mathrm{fv}((\tau_1 \to \tau_2))}) \tag{3.2}$$

Hereby α varies through type variables in *TVar* and τ_1, τ_2 both refer to the non-terminal symbol τ of the grammar.[1] As indicated below the μ-expression in the above defining grammar for *can-μTp*, a type $(\mu\beta.\, \sigma)$ can only be a recursive type in canonical form, if $\beta \in \mathrm{fv}(\sigma)$, i.e., if β has indeed at least

[1] τ_1 and τ_2 have been used here as differently indexed symbols in order to make the form of the parsed expressions more readily understandable in the case of a leading μ-symbol where a side-condition on the variables occurring free comes into play.

one free occurrence in σ, and if σ is furthermore of the form $(\sigma_1 \to \sigma_2)$ for some $\sigma_1, \sigma_2 \in \mathit{can}\text{-}\mu Tp$.

(iii) The *(syntactical) depth* $|\tau|$ of a recursive type $\tau \in \mu Tp$ or $\tau \in \mathit{can}\text{-}\mu Tp$ is a natural number or zero that is defined by induction on the formal structure of τ (described by the grammar in (i)) using the clauses:

$$|\bot| =_{\text{def}} 0 \,, \qquad |\top| =_{\text{def}} 0 \,, \qquad |\alpha| =_{\text{def}} 0 \;\; (\text{for all } \alpha \in \mathit{TVar}) \,,$$
$$|\tau_1 \to \tau_2| =_{\text{def}} 1 + \max\{|\tau_1|, |\tau_2|\} \;\; (\text{for all } \tau_1, \tau_2 \in \mu Tp) \,,$$
$$|\mu\alpha.\, \tau_0| =_{\text{def}} 1 + |\tau_0| \;\; (\text{for all } \alpha \in \mathit{TVar} \text{ and } \tau_0 \in \mu Tp) \,.$$

The *size* or *(syntactical) length* $\mathrm{s}(\tau)$ of a recursive type $\tau \in \mu Tp$ is a natural number that is defined by induction with the following clauses:

$$\mathrm{s}(\bot) =_{\text{def}} 1 \,, \qquad \mathrm{s}(\top) =_{\text{def}} 1 \,, \qquad \mathrm{s}(\alpha) =_{\text{def}} 1 \;\; (\text{for all } \alpha \in \mathit{TVar}) \,,$$
$$\mathrm{s}(\tau_1 \to \tau_2) =_{\text{def}} \mathrm{s}(\tau_1) + \mathrm{s}(\tau_2) \;\; (\text{for all } \tau_1, \tau_2 \in \mu Tp) \,,$$
$$\mathrm{s}(\mu\alpha.\, \tau_0) =_{\text{def}} 1 + \mathrm{s}(\tau_0) \;\; (\text{for all } \alpha \in \mathit{TVar} \text{ and } \tau_0 \in \mu Tp) \,.$$

⊠

With respect to a semantics for recursive types as labeled trees (that will be introduced in Section 3.5) recursive types in canonical form can be seen to describe the labeled trees they represent in a more concise way than recursive types that are not (recursive types) in canonical form. But it will be shown (in Section 3.8) that every recursive type τ in μTp can effectively be transformed into a recursive type τ^c in $\mathit{can}\text{-}\mu Tp$ such that τ and τ^c represent the same labeled tree; on the way from τ to τ^c a number of syntactical redundancies are removed concerning the way how τ describes the labeled tree it denotes.

In the context of the investigation here later of relationships between known proof systems for the equivalence relation "recursive type equality" on recursive types, the distinction between the set μTp and its subset $\mathit{can}\text{-}\mu Tp$ is relevant for the following reason: one axiom system, due to Amadio and Cardelli, is formulated for equations between recursive types in μTp, whereas another, given by Brandt and Henglein, is based on equations between recursive types in canonical from the set $\mathit{can}\text{-}\mu Tp$. But while it can be of considerable technical convenience to let a proof system be based on formulas containing exclusively recursive types in canonical form (this is the case for the system of Brandt and Henglein), there is in general no intrinsic need to do so. We will base all proof systems that are studied here on equations between recursive types in μTp (see the systems in Section 3.7, and especially in Chapter 5). In particular, we will extend the system of Brandt and Henglein in a straightforward way to one that axiomatizes the notion "recursive type equivalence" between all recursive types in μTp.

Agreement 3.1.2 (On the dropping of outermost parentheses and the use of shorthands for recursive types). In the following outermost parentheses of recursive types or recursive types in canonical form are dropped as good as always;

this means that, for instance, in place of the recursive type $(\mu\alpha.\,(\alpha \to (\alpha \to \alpha)))$ in *can-μTp* just $\mu\alpha.\,(\alpha \to (\alpha \to \alpha))$ will be written.

Furthermore, for arbitrary variables $\alpha_1, \ldots, \alpha_n \in \mathit{TVar}$ and recursive types $\tau \in \mu\mathit{Tp}$, the notation $\mu\alpha_1\alpha_2\ldots\alpha_n.\,\tau$ shall be allowed to be used as a shorthand for the recursive type $(\mu\alpha_1.\,(\mu\alpha_2.\,(\ldots(\mu\alpha_n.\,\tau)\ldots)))$.

The countably infinite set *TVar* of type variables will be considered as fixed together with an implicit numbering $\mathit{TVar} = \{\alpha_1, \alpha_2, \alpha_3, \ldots\}$ and will be referred to explicitly only at a few places from now on. Small Greek letters $\alpha, \beta, \gamma, \delta, \epsilon$ (possibly indexed, primed, barred or etc.) will be used as syntactical variables, which vary through type variables; letters $\tau, \sigma, \rho, \omega, \chi$ (again possibly indexed, primed, etc.) will be used as syntactical variables, which vary through recursive types (it will have to be made clear in the context, whether types in *μTp* or in *can-μTp* are meant).

Recursive Types τ and τ', that are *variants* of each other, i.e. that can be transformed into one another by a finite sequence of admissible renamings of bound variables, are *not* identified from the outset here (as this is done so quite frequently in the literature). In particular, we do not introduce here, neither explicitly nor as an implicit convention on how to view occurrences of recursive types, the use of equivalence classes modulo renaming of bound type variables ("α-conversion" equivalence classes on recursive types). For some comments on the reason for this, see the start for Section 3.4. As a consequence of this decision, we have to care for bound-variable renaming explicitly; formal notions and notations concerning the variant relation between recursive types will be introduced in Section 3.4. Furthermore, also substitution in recursive types will be approached rather explicitly here by the definition of the notion "admissible substitution expression" in Section 3.3.

3.2 Contexts, Positions and Subterms

At some occasions we will rely on the following definition of contexts for recursive types. Informally, a "context for a recursive type" can be understood as the result of replacing in a recursive type precisely one variable symbol, or one symbol $\bot$ or $\top$, by a hole $\Box$.

Definition 3.2.1 (Contexts for recursive types). The set *μTp–Ctxt* of *contexts for recursive types* is generated by the following grammar in informal Backus-Naur-Form

$$C ::= \Box \mid (C_1 \to \tau_2) \mid (\tau_1 \to C_2) \mid (\mu\alpha.\,C_0) \tag{3.3}$$

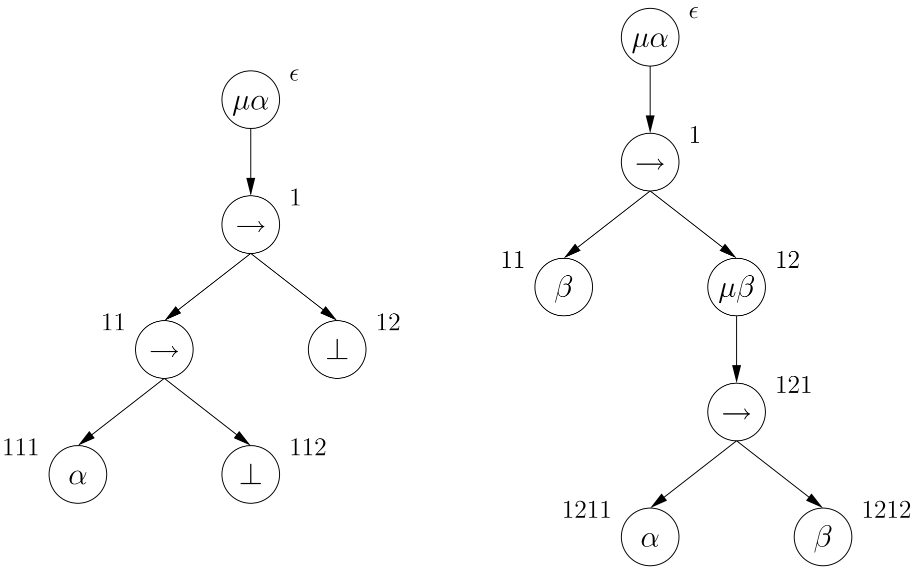
where $\tau_1, \tau_2 \in \mu\mathit{Tp}$, $\alpha \in \mathit{TVar}$ and where we use C as a syntactical variable over contexts. The elements of *μTp–Ctxt* are called *contexts for recursive types*.

The *(syntactical) depth* $|C|$ and *the (syntactical) size* $s(C)$ of a context for recursive types $C \in \mu\mathit{Tp}\text{–}\mathit{Ctxt}$ can be defined inductively by adding the base cases

$$|\Box| =_{\text{def}} 0\,, \qquad\qquad \text{and} \qquad\qquad \mathrm{s}(\Box) =_{\text{def}} 1$$

to the respective five clauses in the inductive definitions of the depth and the size of a recursive type in Definition 3.1.1, (iii).

Figure 3.1: Illustration of the positions in the two recursive types in canonical form $\mu\alpha.\,((\alpha \to \bot) \to \bot)$ and $\mu\alpha.\,(\beta \to \mu\beta.\,(\alpha \to \beta))$.



Hole-filling in contexts for recursive types is defined in the obvious way: for all $C \in \mu Tp\text{–}Ctxt$ and all $\tau \in \mu Tp$, the recursive type $C[\tau]$ is defined by replacing the (single) occurrence of $\Box$ in C by τ. ⊠

In the two following definitions we formally define what we will mean by the set of “positions” of a recursive type, and by the “subterm” of a recursive type τ at some given position in τ.

Definition 3.2.2 (Positions in a recursive type). We define the function

$$Pos : \mu Tp \longrightarrow \mathcal{P}(\{1,2\}^*)$$

that assigns to every recursive type $\tau \in \mu Tp$ the set $Pos(\tau)$ of *positions of* τ by induction on the depth $|\tau|$ of τ, using the following clauses:

$$Pos(\tau) =_{\text{def}} \begin{cases} \{\epsilon\} & \text{if } \tau \equiv \bot, \text{ or } \tau \equiv \top, \\ & \quad \text{or } \tau \equiv \alpha \text{ for some } \alpha \in TVar \\ \{\epsilon\} \cup \{\, i.p \mid i \in \{1,2\},\, p \in Pos(\tau_i)\} & \\ & \text{if } \tau \equiv \tau_1 \to \tau_2 \text{ for some } \tau_1, \tau_2 \in \mu Tp \\ \{\epsilon\} \cup \{\, 1.p \mid p \in Pos(\tau_0)\} & \text{if } \tau \equiv \mu\alpha.\,\tau_0 \text{ for } \alpha \in TVar\,,\ \tau_0 \in \mu Tp \end{cases}$$

⊠

We refer to Figure 3.1 for an illustration of the set of positions in the two recursive types $\tau \equiv \mu\alpha.\,((\alpha \to \bot) \to \bot)$ and $\sigma \equiv (\beta \to \mu\beta.\,(\alpha \to \beta))$. There, the positions of the recursive types τ and σ are associated with nodes in the 'formation trees' of the μ-terms τ and σ, respectively.

Definition 3.2.3 (Subterms of a recursive type). The partial function

$$\begin{aligned} \cdot|. \;:\; \mu Tp \times \{1,2\}^* &\rightharpoonup \mu Tp \\ \langle \sigma, p \rangle &\mapsto \sigma|_p \;, \end{aligned}$$

which assigns to every recursive type $\sigma \in \mu Tp$ and to every position $p \in Pos(\sigma)$ the *subterm* $\sigma|_p$ *of* σ *at position* p (and which is undefined for all other positions), is defined by induction on the length $|p|$ of p by using the following clauses: for all $i \in \{1,2\}$ and $p_0 \in \{1,2\}^*$, we let

$$\sigma|_\epsilon =_{\text{def}} \sigma \quad (\text{for all } \sigma \in \mu Tp) \;,$$
$$(\bot|_{i.p_0}) \uparrow \;, \qquad (\top|_{i.p_0}) \uparrow \;, \qquad (\alpha|_{i.p_0}) \uparrow \quad (\text{for all } \alpha \in TVar) \;,$$
$$(\sigma_1 \to \sigma_2)|_{i.p_0} =_{\text{def}} \sigma_i|_{p_0} \quad (\text{for all } \sigma_1, \sigma_2 \in \mu Tp) \;,$$
$$(\mu\alpha.\,\sigma_0)|_{1.p_0} =_{\text{def}} \sigma_0|_{p_0} \;, \quad ((\mu\alpha.\,\sigma_0)|_{2.p_0}) \uparrow \quad (\text{for all } \alpha \in TVar \text{ and } \sigma_0 \in \mu Tp) \;.$$

Let $\sigma \in \mu Tp$. A recursive type τ is a *subterm of* σ (symbolically denoted by $\tau \trianglelefteq \sigma$) if and only if there exists $p \in Pos(\sigma)$ such that $\tau = \sigma|_p$ holds, i.e. such that τ is the subterm of σ at position p. And we denote by

$$\text{Subt}(\sigma) =_{\text{def}} \{\tau \in \mu Tp \mid \tau \trianglelefteq \sigma\}$$

the *set of subterms of* σ. ⊠

The proposition below formulates the easy provable statement that the number of positions in a recursive type τ equals the size of τ.

Proposition 3.2.4. *For all* $\tau \in \mu Tp$ *it holds that* $|Pos(\tau)| = s(\tau)$.

The main statement of the subsequent proposition is that the definition above of the subterms of a recursive type σ, which hinges on the set of positions in σ, coincides with an obvious recursive definition of what is meant by a "subterm of σ": an "immediate subterm of σ" or a "subterm of an 'immediate subterm of σ'".

Proposition 3.2.5. *Let* $\sigma, \sigma_0, \sigma_1, \sigma_2 \in \mu Tp$ *and* $\beta \in TVar$. *Then the following three statements hold for the set* $Subt(\sigma)$ *of subterms of* σ *:*

(i) $Subt(\sigma) = \{\sigma\}$ *for all* $\sigma \in \{\bot, \top\} \cup TVar$.

(ii) $Subt(\sigma_1 \to \sigma_2) = \{\sigma_1 \to \sigma_2\} \cup Subt(\sigma_1) \cup Subt(\sigma_2)$.

(iii) $Subt(\mu\beta.\,\sigma) = \{\mu\beta.\,\sigma\} \cup Subt(\sigma)$.

And furthermore it holds that

$$|Subt(\sigma)| \leq s(\sigma) \ ,$$

i.e. that the number of subterms of a recursive type σ is bounded by the size of σ.

And furthermore we define, for all recursive types τ, the set of positions in τ of subterms that start with a μ-binding.

Definition 3.2.6 (Subterms starting with a μ-binding in a recursive type). We define the functions

$$\begin{aligned} \mu Pos : \mu Tp &\longrightarrow \mathcal{P}(\{1,2\}^*) \\ \tau &\longmapsto \mu Pos(\tau) =_{\text{def}} \big\{ p \in Pos(\tau) \ \mid \ \tau|_p = \mu\beta.\,\rho, \ \beta \in TVar, \ \rho \in \mu Tp \big\} \ , \\ \mu Subt : \mu Tp &\longrightarrow \mu Tp \\ \tau &\longmapsto \mu Subt(\tau) =_{\text{def}} \big\{ \sigma \mid \sigma \in TVar, \ (\exists\, p \in \mu Pos(\tau))\, [\, \sigma = \tau|_p \,] \big\} \ , \end{aligned}$$

which assign to every recursive type τ the set of all positions p in τ such that $\tau|_p$ starts with a μ-binding, and respectively, the set of all subterms of τ that start with a μ-binding. ⊠

3.3 Substitution Expressions

In this section we describe how we will formally treat expressions involving the substitution operation on recursive types. We have chosen not to define substitution as a total operation in the style of Curry's definition of the substitution operation in λ-calculus (for example, see [HS86, Def. 1.11, p.7]). Instead, we will admit an expression involving a substitution operation between recursive types *only if* the substitution can be carried out as a direct replacement that does not lead to unwanted bindings of free variables from the substituted terms. This approach leads us to a formal definition of "substitution expressions involving recursive types" (in Definition 3.3.1), and to a stipulation of when a substitution expression is "admissible" as well as of which recursive type, if any, "is denoted" by an "admissible" substitution expression (in Definition 3.3.2). Eventually, we are being led to the adoption of an implicit side-condition on the occurrence of a substitution expressions (in Convention 3.3.6): basically, later only "admissible" substitution expressions will be allowed to occur.

For the purpose of describing and justifying such transformation-steps between substitution expressions that preserve admissibility and the denoted recursive types, we introduce the notion of "equality implications" between substitution expressions (in Definition 3.3.8). These are formal statements about two substitution expressions, asserting that the implicit side-condition on one substitution expression entails the implicit side-condition on the other and that both of them denote the same recursive type. In a similar way, also "equality equivalences" will be defined. Furthermore, we give two basic lemmas that contain conditions under which easy kind of transformations between substitution expressions are possible. The second one of these lemmas is concerned with 'commuting' an application of the substitution operation over another such application. Finally, and at first sight not related to substitution expressions, we define three "variable conditions" for recursive types that will play a role in proving a theorem in Section 3.9.

The occurrence of a "substitution expression" of the form $\tau[\sigma/\alpha]$ will, apart from certain exceptional cases mentioned below in Convention 3.3.6, later (that is, from Convention 3.3.6 onwards) always be subject to the *implicit side-condition* on τ, σ and α that the substitution of σ for α in τ can be performed directly. By this we mean that it can be carried out as the mere replacement of all free occurrences of α in τ by σ without giving rise to *unwanted bindings*[2], i.e. bindings of free variables of σ by binders within τ. In this way the occurrence of a substitution expression $\tau[\sigma/\alpha]$ is restricted to the case in which it is not necessary to rename bound variables of τ to guarantee that during the process of inserting σ for the free occurrences of α in τ bindings of free variables in σ are avoided. A consequence is that, for given $\tau, \sigma \in \mu Tp$ and $\alpha \in TVar$, generally renamings of bound variables in τ have to be cared for, with the result of a recursive type τ' that differs from τ only by the names of (some, possibly all, of its) bound variables, to make the use of a substitution expression $\tau'[\sigma/\alpha]$ possible.

Let $\tau, \sigma \in \mu Tp$ and $\alpha \in TVar$. We say that σ *is substitutible for* α *in* τ if and only if it holds for every variable $\beta \in \mathrm{fv}(\sigma)$ that there does not exist a free occurrence of α in τ within a subterm of the form $\mu\beta.\,\rho$ of τ.[3]

In the following definition, we fix a notion of "substitution expression" involving recursive types by giving a grammar that generalizes the defining grammar for recursive types: in addition to the productions of grammar (3.1), "substitution expressions" may also be formed, and typically will be formed, by applications of a ternary operator $(\cdot)[\cdot/\cdot]$ that is used to symbolize single substitutions, and/or by applications of operators $(\cdot)[\cdot/\cdot, \ldots, \cdot/\cdot]$ with arity $(2n+1)$ (for $n \in \omega$, $n \geq 2$) that are used to symbolize n parallel substitutions.

Definition 3.3.1 (Substitution Expressions). Let $TVar$ be an infinite set of type variables. Then the set $SubstExpr(TVar)$ of *substitution expressions involving recursive types* (for short, the set of *substitution expressions*) on $TVar$ is generated by the following grammar:

$$\begin{aligned} \mathbf{s} \;::=\; \bot \mid \top \mid \alpha \mid \mathbf{s} \to \mathbf{s} \mid \mu\alpha.\,\mathbf{s} \mid \mathbf{s}[\mathbf{s}/\alpha] \mid \mathbf{s}[\mathbf{s}/\alpha_1, \ldots, \mathbf{s}/\alpha_n] \\ \text{(where } \alpha \in TVar,\ n \in \omega\backslash\{0\},\ \text{and } \alpha_1, \ldots, \alpha_n \in TVar) \;. \end{aligned} \tag{3.4}$$

Here and later, we use $\mathbf{s}$ and $\mathbf{t}$, possibly indexed and/or with attached accents, stars, etc., as syntactical variables that vary through substitution expressions. We will usually write $SubstExpr$ for the set $SubstExpr(TVar)$ of substitution expressions

[2] For the undesirable situation of bindings that are able to distort the intended meaning of a substitution sometimes also the expression "clashes of variables" is used.

[3] By using terminology that also occurs frequently in the literature, the condition "σ is substitutible for α in τ" could, equally well, be referred to as "τ is free for α in σ".

on *TVar*, and in doing so, we leave the set of type variables implicit on which *SubstExpr* depends.

We call a substitution expression $\mathbf{s}$ *trivial* if and only if $\mathbf{s}$ does not contain substitution operators $[\cdot/\cdot]$ nor $[\cdot/\cdot, \ldots, \cdot/\cdot]$ at all, or more precisely, if during the generation of $\mathbf{s}$ according to the grammar (3.4) the sixth and the seventh production are never employed. ⊠

From the definition of recursive types by grammar (3.1), and the definition of substitution expressions by grammar (3.4), it obviously follows that a substitution expression is trivial if and only if it is a recursive type.

Let $\tau, \sigma \in \mu Tp$ and $\alpha \in TVar$, and let $\mathbf{s}$ be the substitution expression $\tau[\sigma/\alpha]$. We mentioned above that a substitution expression like $\mathbf{s}$ will later only be admitted, apart from definite exceptions, if the substitution it formalizes can be carried out without leading to unwanted bindings: in the case of $\mathbf{s}$ this means that $\mathbf{s}$ will be "admissible" if and only if σ is substitutible for α in τ. In the definition below this notion of "admissibility" for basic substitution expressions such as $\mathbf{s}$ is extended to the set of all substitution expressions in a straightforward way. For auxiliary purposes, and as a related notion of obvious interest, we also define, for all substitution expressions $\mathbf{t}$, the recursive type that "is denoted" by $\mathbf{t}$ in case that $\mathbf{t}$ is "admissible". With regard to this notion of "denoted recursive type", the definition below generalizes the stipulation suggested by the example of the substitution expression $\mathbf{s} \equiv \tau[\sigma/\alpha]$: if $\mathbf{s}$ is "admissible", then $\mathbf{s}$ "denotes" the result of replacing all free occurrences of α in τ by σ.

Definition 3.3.2 (Admissible substitution expressions; the recursive type that is denoted by a substitution expression). For all substitution expressions $\mathbf{s} \in SubstExpr$, we define when $\mathbf{s}$ is called *admissible*: this is done, using the clauses (i)–(v) below, by induction on the formation of $\mathbf{s}$ according to the grammar (3.4). In parallel with admissibility of substitution expressions, also the partial function

$$[\![\cdot]\!] \;:\; SubstExpr \rightharpoonup \mu Tp$$

is (and needs to be) defined that to every substitution expression $\mathbf{s}$ that is admissible assigns the recursive type $[\![\mathbf{s}]\!]$ that *is denoted by* $\mathbf{s}$ (we also will say that $\mathbf{s}$ *denotes* $[\![\mathbf{s}]\!]$).

(i) Suppose that $\mathbf{s} \in \{\bot, \top\} \cup TVar$. Then we stipulate that $\mathbf{s}$ is admissible; and we agree that $\mathbf{s}$ denotes itself (the recursive type $\mathbf{s}$), i.e. we let $[\![\mathbf{s}]\!] =_{\mathrm{def}} \mathbf{s}$.

(ii) Suppose that $\mathbf{s} \equiv \mathbf{s}_1 \to \mathbf{s}_2$ for some $\mathbf{s}_1, \mathbf{s}_2 \in SubstExpr$. Then we stipulate that $\mathbf{s}_1 \to \mathbf{s}_2$ is admissible if and only if both $\mathbf{s}_1$ and $\mathbf{s}_2$ are admissible. If $\mathbf{s}$ is indeed admissible, the recursive type denoted by $\mathbf{s}_1 \to \mathbf{s}_2$ is defined by $[\![\mathbf{s}]\!] =_{\mathrm{def}} [\![\mathbf{s}_1]\!] \to [\![\mathbf{s}_2]\!]$; otherwise $[\![\mathbf{s}]\!]$ is undefined.

(iii) Suppose that $\mathbf{s} \equiv \mu\alpha.\, \mathbf{s}_0$ for some substitution expression $\mathbf{s}_0 \in SubstExpr$ and some type variable $\alpha \in TVar$. We say that $\mathbf{s}$ is admissible if and only if $\mathbf{s}_0$ is admissible. And if $\mathbf{s}$ is admissible, then it denotes the recursive type $[\![\mathbf{s}]\!] =_{\mathrm{def}} \mu\alpha.\, [\![\mathbf{s}_0]\!]$; otherwise we let $[\![\mathbf{s}]\!]\!\uparrow$.

(iv) Suppose that $\mathbf{s} \equiv \mathbf{s}_0[\mathbf{s}_1/\alpha]$ for some $\alpha \in \mathit{TVar}$ and some $\mathbf{s}_0, \mathbf{s}_1 \in \mathit{SubstExpr}$. Then we say that $\mathbf{s}$ is admissible if and only if $\mathbf{s}_0$ and $\mathbf{s}_1$ are admissible and if additionally $[\![\mathbf{s}_1]\!]$ is substitutible for α in $[\![\mathbf{s}_0]\!]$. If $\mathbf{s}$ is indeed admissible, then we let the recursive type $[\![\mathbf{s}]\!]$ denoted by $\mathbf{s}$ be the result of replacing all free occurrences of α in the recursive type $[\![\mathbf{s}_0]\!]$ by the recursive type $[\![\mathbf{s}_1]\!]$; it $\mathbf{s}$ is not admissible, then $[\![\mathbf{s}]\!]$ is undefined.

(v) Suppose that $\mathbf{s} \equiv \mathbf{s}_0[\mathbf{s}_1/\alpha_1, \ldots, \mathbf{s}_n/\alpha_n]$ for some $n \in \omega \setminus \{0, 1\}$, $\alpha_1, \ldots, \alpha_n \in$ $\in \mathit{TVar}$, and $\mathbf{s}_1, \ldots, \mathbf{s}_n \in \mathit{SubstExpr}$. Then we call $\mathbf{s}$ admissible if and only if the following three conditions hold: (1) for all $i, j \in \{1, \ldots, n\}$ such that $i \neq j$, $\alpha_i \not\equiv \alpha_j$ holds, (2) the substitution expressions $\mathbf{s}_0$, $\mathbf{s}_1$, …, $\mathbf{s}_n$ are admissible, and (3) for all $i \in \{1, \ldots, n\}$, $[\![\mathbf{s}_i]\!]$ is substitutible for α in $[\![\mathbf{s}_0]\!]$. And if $\mathbf{s}$ is indeed admissible, then the recursive type $[\![\mathbf{s}]\!]$ denoted by $\mathbf{s}$ is defined as the result of simultaneously replacing the free occurrences of $\alpha_1, \ldots, \alpha_n$ in $[\![\mathbf{s}_0]\!]$ by $[\![\mathbf{s}_1]\!]$, …, $[\![\mathbf{s}_n]\!]$, respectively. If $\mathbf{s}$ is not admissible, then we let $[\![\mathbf{s}]\!]$ be undefined. ⊠

The proposition below formulates the obvious consequence of Definition 3.3.2 that a substitution expression is admissible if and only if it denotes a recursive type.

Proposition 3.3.3. *For all substitution expressions $\mathbf{s} \in \mathit{SubstExpr}$ it holds:*

$$\mathbf{s} \textit{ is admissible} \iff [\![\mathbf{s}]\!]\!\downarrow \;\;\&\;\; [\![\mathbf{s}]\!] \in \mu\mathit{Tp}\,.$$

As an illustration of the notions introduced in Definition 3.3.2, we consider substitution expressions which are of a form that plays a role in Lemma 3.3.11 below.

Example 3.3.4. Let $\tau, \sigma_1, \sigma_2 \in \mu\mathit{Tp}$ and $\alpha, \beta \in \mathit{TVar}$. We consider the substitution expression

$$\mathbf{s} \equiv \tau\,[\sigma_2/\beta]\,[\sigma_1[\sigma_2/\beta]/\alpha]\,. \tag{3.5}$$

By Definition 3.3.2, we find that $\mathbf{s}$ is admissible if and only if the following three conditions are fulfilled:

[C1] σ_2 is substitutible for β in τ;

[C2] σ_2 is substitutible for β in σ_1;

[C3] the recursive type $[\![\sigma_1[\sigma_2/\beta]]\!]$ that is denoted by (the admissible substitution expression) $\sigma_1[\sigma_2/\beta]$ is substitutible for α in the recursive type $[\![\tau[\sigma_2/\beta]]\!]$ that is denoted by (the admissible substitution expression) $\tau[\sigma_2/\beta]$.

Given that the conditions [C1], [C2], and [C3] hold, the substitution expression $\mathbf{s}$ denotes the recursive type $\tilde{\tau}$ that is the result of replacing all free occurrences of α in $[\![\tau[\sigma_2/\beta]]\!]$ by $[\![\sigma_1[\sigma_2/\beta]]\!]$; in this case we have

$$[\![\mathbf{s}]\!] = \Big[\!\!\Big[\, \tau[\sigma_2/\beta]\,\big[\sigma_1[\sigma_2/\beta]/\alpha\big] \,\Big]\!\!\Big] = \Big[\!\!\Big[\, [\![\tau[\sigma_2/\beta]]\!]\,\big[[\![\sigma_1[\sigma_2/\beta]]\!]/\alpha\big] \,\Big]\!\!\Big] = \tilde{\tau}\,.$$

■

Now we define what will be meant by the "implicit side-condition" of a substitution expression, and subsequently we state the convention to which we will adhere concerning the use of substitution expressions.

Definition 3.3.5 (Implicit side-condition on a substitution expression). Let $\mathbf{s}$ be a substitution expression. By the *implicit side-condition on* $\mathbf{s}$ we mean the condition that $\mathbf{s}$ is admissible. ⊠

Convention 3.3.6 (The occurrence of substitution expressions). In this and in all later chapters, we will use the following convention on the occurrence of substitution expressions: with the exception of the situations described in items (Exc1) and (Exc2) below, we allow a substitution expression $\mathbf{s}$ to occur only if the implicit side-condition on $\mathbf{s}$ is fulfilled. The two kinds of exceptions to this convention are:

(Exc1) If in a certain context there is the need to state explicitly for a considered substitution expression $\mathbf{s}$ that the implicit side-condition on $\mathbf{s}$ is *not* satisfied, then the expression $\mathbf{s}$ will be allowed to occur in a sentence of a form like "$\mathbf{s}$ is not an admissible substitution expression" or "the implicit side-condition on $\mathbf{s}$ is not satisfied".

(Exc2) The second kind of exceptions regards statements of equality implications between substitution expressions in the sense of Definition 3.3.8 below.

⊠

Remark 3.3.7. The reason for exception (Exc2) in item (i) of Convention 3.3.6 on the use of substitution expressions is, informally, the following. The meaning of an "equality implication" $\mathbf{s} \Rrightarrow \mathbf{t}$, for some substitution expressions $\mathbf{s}$ and $\mathbf{t}$, will be defined as the assertion that admissibility of $\mathbf{s}$ implies admissibility of $\mathbf{t}$, and that $\mathbf{s}$ and $\mathbf{t}$ denote the same recursive type. Hence in such statements an interdependence between the implicit side-conditions on two different substitution expressions is asserted and therefore these conditions are not imposed from the outset (otherwise such "equality implication" statements would all be true trivially).

For the practical treatment of substitution expressions, we need to be able to carry out a number of such simple transformations on substitution expressions that preserve admissibility and the denoted recursive types. As a formalization of statements that justify such transformations, we introduce "equality implications" and "equality equivalences" between substitution expressions. Hereby a substitution expression $\mathbf{s}$ is understood to 'syntactically imply' another substitution expression $\mathbf{t}$ if and only if (i) the implicit side-condition on $\mathbf{s}$ implies the implicit side-condition on $\mathbf{t}$ and (ii) the recursive types denoted by $\mathbf{s}$ and $\mathbf{t}$ are syntactically equal. This is what is stipulated by the following definition.

Definition 3.3.8 (Equality implications and equality equivalences). Formally, an *equality implication* is an expression $\mathbf{s} \Rrightarrow \mathbf{t}$ or $\mathbf{s} \Lleftarrow \mathbf{t}$, where $\mathbf{s}$ and $\mathbf{t}$ are substitution expressions. An *equality equivalence* is an expression $\mathbf{s} \Leftarrow\!\!\!\Rightarrow \mathbf{t}$ for two substitution expressions $\mathbf{s}$ and $\mathbf{t}$.

Let $\mathbf{s}$ and $\mathbf{t}$ be substitution expressions. We say that *the equality implication* $\mathbf{s} \Rrightarrow \mathbf{t}$ *holds* if and only if

$$\mathbf{s} \text{ is admissible } \Longrightarrow \mathbf{t} \text{ is admissible } \& \ [\![\mathbf{s}]\!] = [\![\mathbf{t}]\!]$$

holds. We say that $\mathbf{s} \Lleftarrow \mathbf{t}$ *holds* if and only if $\mathbf{t} \Rrightarrow \mathbf{s}$ holds. And we stipulate that *the equality equivalence* $\mathbf{s} \Leftrightarrow \mathbf{t}$ *holds* iff both of $\mathbf{s} \Rrightarrow \mathbf{t}$ and $\mathbf{s} \Lleftarrow \mathbf{t}$ hold, or equivalently, iff

$$\mathbf{s} \text{ is admissible } \vee \ \mathbf{t} \text{ is admissible } \Longrightarrow \mathbf{s} \text{ and } \mathbf{t} \text{ are admissible } \& \ [\![\mathbf{s}]\!] = [\![\mathbf{t}]\!]$$

holds. ☒

We give the following example with the aim of illustrating Definition 3.3.8.

Example 3.3.9. Let $\tau, \sigma_1, \sigma_2 \in \mu Tp$ and $\alpha, \beta \in TVar$. We consider the equality implication

$$\tau[\sigma_1/\alpha]\,[\sigma_2/\beta] \ \Rrightarrow \ \tau[\sigma_2/\beta]\,[\sigma_1[\sigma_2/\beta]/\alpha] \ , \tag{3.6}$$

which occurs in a lemma below. By spelling out, rather explicitly, the conditions for this statement to hold, we find: (3.6) holds if and only if the following statement is true:

> Suppose, that the substitution expression on the left side of (3.6) is admissible, which means: σ_1 is substitutible for α in τ, and that σ_2 is substitutible for β in the recursive type $[\![\tau[\sigma_1/\alpha]]\!]$ that is denoted by (the admissible substitution expression) $\tau[\sigma_1/\alpha]$. Then it follows, that:
>
> - the substitution expression on the left side in (3.6) is admissible, which means: σ_2 is substitutible for β in τ, σ_2 is substitutible for β in σ_1, and the recursive type $[\![\sigma_1[\sigma_2/\beta]]\!]$ that is denoted by (the admissible substitution expression) $\sigma_1[\sigma_2/\beta]$ is substitutible for α in $[\![\tau[\sigma_2/\beta]]\!]$, the recursive type that is denoted by (the admissible substitution expression) $\tau[\sigma_2/\beta]$; and
> - the substitution expressions on either side of (3.6) denote the same recursive type: in particular it holds that the recursive types denoted by (the two admissible substitution expressions) $[\![\tau[\sigma_1/\alpha]]\!]\,[\sigma_2/\beta]$ and $[\![\tau[\sigma_2/\beta]]\!]\,[[\![\sigma_1[\sigma_2/\beta]]\!]/\alpha]$ are syntactically equal; it follows that
>
> $$\begin{aligned}[\![\,\tau[\sigma_1/\alpha]\,[\sigma_2/\beta]\,]\!] \ = \ [\![\,[\![\tau[\sigma_1/\alpha]]\!]\,[\sigma_2/\beta]\,]\!] \ &= \ [\![\,[\![\tau[\sigma_2/\beta]]\!]\,\big[\,[\![\sigma_1[\sigma_2/\beta]]\!]/\alpha\,\big]\,]\!] \\ &= \ [\![\,\tau[\sigma_2/\beta]\,[\sigma_1[\sigma_2/\beta]/\alpha]\,]\!]\end{aligned}$$
>
> is the case. ■

As a first useful statement for the purpose of transforming substitution expressions, we give the following lemma.

Lemma 3.3.10. *Let $\tau, \sigma \in \mu Tp$ and α, β be different type variables, i.e. $\alpha \not\equiv \beta$. Then the following three statements are true:*

(i) $(\mu\beta.\,\tau_0)[\sigma/\alpha] \Rrightarrow \mu\beta.\,\tau_0[\sigma/\alpha]$.

(ii) If $\alpha \notin fv(\tau_0)$ *or* $\beta \notin fv(\sigma)$*, then* $(\mu\beta.\,\tau_0)[\sigma/\alpha] \Lleftarrow \mu\beta.\,\tau_0[\sigma/\alpha]$.

(iii) If $\alpha \notin fv(\tau_0)$ *or* $\beta \notin fv(\sigma)$*, then* $(\mu\beta.\,\tau_0)[\sigma/\alpha] \Leftrightarrow \mu\beta.\,\tau_0[\sigma/\alpha]$.

Proof. We will only prove item (ii) of the lemma because item (i) can be shown analogously and in an easier way, and because item (iii) is an immediate logical consequence of the assertions in (i) and (ii). But we start with the following general observation, which is the key to the proofs of all three items.

Let $\tau, \sigma \in \mu Tp$ and $\alpha, \beta \in TVar$ such that $\alpha \not\equiv \beta$. Then the substitution expression $(\mu\beta.\,\tau_0)[\sigma/\alpha]$ is admissible if and only if it holds that (a) $\tau_0[\sigma/\alpha]$ is admissible and that (b) if $\alpha \in \mathrm{fv}(\tau_0)$ then also $\beta \notin \mathrm{fv}(\sigma)$ is the case. Hence the admissibility of $(\mu\beta.\,\tau_0)[\sigma/\alpha]$ entails the admissibility of $\mu\beta.\,\tau_0[\sigma/\alpha]$, whereas for $(\mu\beta.\,\tau_0)[\sigma/\alpha]$ to be admissible also the condition $\alpha \notin \mathrm{fv}(\tau_0) \vee \beta \notin \mathrm{fv}(\sigma)$ is needed additionally to the assumption that $\mu\beta.\,\tau_0[\sigma/\alpha]$ is admissible.

To prove (ii) now, we let $\tau, \sigma \in \mu Tp$ and $\alpha, \beta \in TVar$ such that $\alpha \not\equiv \beta$, and we assume that $\mathbf{s}_2 \equiv \mu\beta.\,\tau_0[\sigma/\alpha]$ is admissible and that $\alpha \notin \mathrm{fv}(\tau_0)$ or $\beta \notin \mathrm{fv}(\sigma)$ holds. Then we have to show that $\mathbf{s}_1 \equiv (\mu\beta.\,\tau_0)[\sigma/\alpha]$ is admissible and that $\mathbf{s}_1$ and $\mathbf{s}_2$ denote the same recursive type. In the preceding paragraph we have observed that the condition $\alpha \notin \mathrm{fv}(\tau_0) \vee \beta \notin \mathrm{fv}(\sigma)$ is sufficient to conclude the admissibility of $\mathbf{s}_1$ from the admissibility of $\mathbf{s}_2$. Since this condition is fulfilled here by assumption, it follows that $\mathbf{s}_1$ is admissible as well. Furthermore $\mathbf{s}_1$ and $\mathbf{s}_2$ denote the same recursive type because, due to $\alpha \not\equiv \beta$, the result of replacing all free occurrences of α in $\mu\beta.\,\tau_0$ is the same as the result of inserting the outcome of replacing all free occurrences of α in τ_0 by σ into the context $(\mu\beta.\,\Box)$.

□

In later proofs we furthermore have to be able to treat the operation of 'commuting' (interchanging) substitutions in a precise way. For this purpose we need an appropriate analogue for a well-known substitution lemma in λ-calculus with respect to our convention on the use of substitution expressions. In particular we consider the question whether, for all $\tau, \sigma_1, \sigma_2 \in \mu Tp$ and $\alpha, \beta \in TVar$ with the properties $\alpha \not\equiv \beta$ and $\alpha \notin \mathrm{fv}(\sigma_2)$, it is true that

$$\tau[\sigma_1/\alpha]\,[\sigma_2/\beta] \;\Leftrightarrow\; \tau[\sigma_2/\beta]\,[\sigma_1[\sigma_2/\beta]/\alpha] \tag{3.7}$$

holds. Perhaps surprisingly it turns out that this is not always the case. Although, under the assumptions $\alpha \not\equiv \beta$ and $\alpha \notin \mathrm{fv}(\tau_2)$, it never happens that the substitution expressions on the left and on the right side of (3.7) are both admissible but denote different recursive types, it is possible that one of them is admissible whereas the other is not. For instance, "$\Rrightarrow$" in the equality equivalence "$\Leftrightarrow$" in (3.7) goes wrong for $\tau, \sigma_1, \sigma_2 \in \mu Tp$ and $\alpha, \beta \in TVar$ with $\alpha \not\equiv \beta$ and $\alpha \notin \mathrm{fv}(\tau_2)$, if it holds that $\alpha \notin \mathrm{fv}(\tau)$, $\tau[\sigma_2/\beta]$ is admissible and σ_2 is not substitutible for β in σ_1. Then $\tau[\sigma_1/\alpha]\,[\sigma_2/\beta]$ is an admissible substitution expression, but $\tau[\sigma_2/\beta]\,[\sigma_1[\sigma_2/\beta]/\alpha]$ is not admissible, because $\sigma_1[\sigma_2/\beta]$ is not admissible. It is easy to give concrete counterexamples that follow this argumentation.

On the other hand, "$\Leftarrow$" in the equivalence "$\Leftrightarrow$" in (3.7) fails for $\tau, \sigma_1, \sigma_2 \in \mu Tp$ and α, β with $\alpha \not\equiv \beta$ and $\alpha \notin \mathrm{fv}(\tau_2)$ in the case that $\tau[\sigma_2/\beta]\,[\sigma_1[\sigma_2/\beta]/\alpha]$ is admissible, but that σ_1 is not substitutible for α in τ (then neither $\tau[\sigma_1/\alpha]$ nor $\tau[\sigma_1/\alpha]\,[\sigma_2/\beta]$ can be admissible). For an example, choose three mutually different variables α, β, and γ, and let $\tau \equiv \mu\beta.\,\alpha$, $\sigma_1 \equiv \beta$, and $\sigma_2 \equiv \gamma$. Then the substitution expression $\tau[\sigma_2/\beta]\,[\sigma_1[\sigma_2/\beta]/\alpha]$ is admissible and it denotes the recursive type $\mu\beta.\,\gamma$, as can be seen in detail from

$$\begin{aligned} \tau[\sigma_2/\beta]\,[\sigma_1[\sigma_2/\beta]/\alpha] \;\equiv\; (\mu\beta.\,\alpha)\,[\gamma/\beta]\,[\beta[\gamma/\beta]/\alpha] \;&\Leftrightarrow\; (\mu\beta.\,\alpha)\,[\beta[\gamma/\beta]/\alpha] \\ &\Leftrightarrow\; (\mu\beta.\,\alpha)\,[\gamma/\alpha] \\ &\Leftrightarrow\; \mu\beta.\,\alpha[\gamma/\alpha] \\ &\Leftrightarrow\; \mu\beta.\,\gamma\,, \end{aligned}$$

(the equality equivalences used above are each of an easily justifiable kind). However, since β is not substitutible for α in $\mu\beta.\,\alpha$, $\tau[\sigma_1/\alpha]$ is not admissible and so neither $\tau[\sigma_1/\alpha]\,[\sigma_2/\beta]$ can be admissible.

As a way to remedy the undesirable situation that the equality implication (3.7) cannot always be relied on, we give the the following lemma. In its statement the two equality implications "$\Rightarrow$" and "$\Leftarrow$" within the equality equivalence (3.7) are considered separately, and conditions are given under which each of them does in fact hold.

Lemma 3.3.11 (Interchanging of Substitutions). *Let $\tau, \sigma_1, \sigma_2 \in \mu Tp$ and $\alpha, \beta \in TVar$ be such that $\alpha \not\equiv \beta$ and $\alpha \notin fv(\sigma_2)$. Then the following three statements are true:*

(i) If $\sigma_1[\sigma_2/\beta]$ is admissible, i.e. if σ_2 is substitutible for β in σ_1, then it holds that

$$\tau[\sigma_1/\alpha]\,[\sigma_2/\beta] \;\Rightarrow\; \tau[\sigma_2/\beta]\,[\sigma_1[\sigma_2/\beta]/\alpha]\,.$$

(ii) If $\beta \notin fv(\sigma_1)$ or if $\tau[\beta/\alpha]$ is admissible (that is, if β is substitutible for α in τ), then it holds that

$$\tau[\sigma_1/\alpha]\,[\sigma_2/\beta] \;\Leftarrow\; \tau[\sigma_2/\beta]\,[\sigma_1[\sigma_2/\beta]/\alpha]\,.$$

(iii) If both of the substitution expressions $\tau[\sigma_1/\alpha]\,[\sigma_2/\beta]$ and $\tau[\sigma_2/\beta]\,[\sigma_1[\sigma_2/\beta]/\alpha]$ are admissible, then they denote the same recursive type.

A *proof* for this lemma is given in Appendix A, Section A.1, on page 333.

Remark 3.3.12. The assumptions in assertions (i) and (ii) of the lemma can be weakened slightly with the result of variant assumptions that may be easier to check in some cases.

In (i), the assumption "$\sigma_1[\sigma_2/\beta]$ is admissible" can be replaced by "$\alpha \in \mathrm{fv}(\tau)$ or $\sigma_1[\sigma_2/\beta]$ is admissible", or equivalently by

$$\alpha \notin \mathrm{fv}(\tau) \;\Longrightarrow\; \sigma_1[\sigma_2/\beta] \text{ is admissible}\,, \tag{3.8}$$

and the resulting assertion is equivalent to (i). This is because if $\alpha \in \mathrm{fv}(\tau)$ holds and if $\tau[\sigma_1/\alpha]\,[\sigma_2/\beta]$ is admissible, then also $\sigma_1[\sigma_2/\beta]$ is admissible.

And in (ii), the assumption "$\beta \notin \mathrm{fv}(\sigma_1)$ or $\tau[\beta/\alpha]$" can be replaced by "$\beta \notin \mathrm{fv}(\sigma_1)$ or $\beta \in \mathrm{fv}(\sigma_2)$ or $\tau[\beta/\alpha]$ is admissible", or equivalently by

$$\beta \in \mathrm{fv}(\sigma_1) \;\;\&\;\; \beta \notin \mathrm{fv}(\sigma_2) \;\;\Longrightarrow\;\; \tau[\beta/\alpha] \text{ is admissible}\;, \tag{3.9}$$

and the resulting assertion is equivalent to (ii) in the lemma. The reason for this is that if $\beta \in \mathrm{fv}(\sigma_2)$ and $\beta \in \mathrm{fv}(\sigma_1)$ holds, then the admissibility of the substitution expression $\tau[\sigma_2/\beta]\,[\sigma_1[\sigma_2/\beta]/\alpha]$ implies that β is substitutible for α in τ. What is more, condition (3.9) can further be weakened formally to the equivalent statement

$$\alpha \in \mathrm{fv}(\tau) \;\;\&\;\; \beta \in \mathrm{fv}(\sigma_1) \;\;\&\;\; \beta \notin \mathrm{fv}(\sigma_2) \;\;\Longrightarrow\;\; \tau[\beta/\alpha] \text{ is admissible}\;. \tag{3.10}$$

(3.10) is equivalent to (3.9), because if $\alpha \notin \mathrm{fv}(\tau)$ is the case then $\tau[\beta/\alpha]$ is admissible.

The following lemma is an easy consequence of the assertions (i) and (ii) in Lemma 3.3.11.

Lemma 3.3.13. *Let $\tau, \sigma_1, \sigma_2 \in \mu Tp$ and $\alpha, \beta \in TVar$.*
If $\alpha \not\equiv \beta$, $\alpha \notin fv(\sigma_2)$, and if one of the two conditions

(i) $\alpha \in fv(\tau)$ and σ_1 is substitutible for α in τ, or

(ii) both of the substitution expressions $\tau[\sigma_1/\alpha]$ and $\sigma_1[\sigma_2/\beta]$ are admissible, i.e. σ_1 is substitutible for α in τ, and σ_2 is substitutible for β in σ_1,

is fulfilled, then it holds that

$$\tau[\sigma_1/\alpha]\,[\sigma_2/\beta] \;\Leftrightarrow\; \tau[\sigma_2/\beta]\,[\sigma_1[\sigma_2/\beta]/\alpha]\;.$$

The following lemma is an easy generalization of Lemma 3.3.11: under some straightforward adaptations of hypotheses, the statements (i), (ii), and (iii) of Lemma 3.3.11 stay correct when the recursive types occurring there are replaced by substitution expressions.

Lemma 3.3.14 (Interchanging of substitutions, general version). *Let $\alpha, \beta \in$ $\in TVar$ and $\mathbf{t}, \mathbf{s}_1, \mathbf{s}_2 \in SubstExpr$ be such that $\alpha \not\equiv \beta$, and that, if $\mathbf{s}_2$ is admissible, also $\alpha \notin fv(\llbracket \mathbf{s}_2 \rrbracket)$ holds. Then the following statements are true:*

(i) If the substitution expression $\mathbf{s}_1[\mathbf{s}_2/\beta]$ is admissible, then it holds that

$$\mathbf{t}[\mathbf{s}_1/\alpha]\,[\mathbf{s}_2/\beta] \;\Rrightarrow\; \mathbf{t}[\mathbf{s}_2/\beta]\,[\mathbf{s}_1[\mathbf{s}_2/\beta]/\alpha]\;.$$

(ii) If in case that $\mathbf{s}_1$ is admissible $\beta \notin fv(\llbracket \mathbf{s}_1 \rrbracket)$ holds, or if $\mathbf{t}[\beta/\alpha]$ is admissible, then it holds that

$$\mathbf{t}[\mathbf{s}_1/\alpha]\,[\mathbf{s}_2/\beta] \;\Lleftarrow\; \mathbf{t}[\mathbf{s}_2/\beta]\,[\mathbf{s}_1[\mathbf{s}_2/\beta]/\alpha]\;.$$

(iii) If both of the substitution expressions $\mathbf{t}[\mathbf{s}_1/\alpha]\,[\mathbf{s}_2/\beta]$ *and* $\mathbf{t}[\mathbf{s}_2/\beta]\,[\mathbf{s}_1[\mathbf{s}_2/\beta]/\alpha]$ *are admissible, then they denote the same recursive type.*

The *proof* of this lemma follows easily from Lemma 3.3.11, using the notions introduced in Definition 3.3.2 and Definition 3.3.8, and therefore it is omitted here.

Concluding this section on substitution expressions, we introduce three variable conditions for recursive types, which form important assumptions for Theorem 3.9.12 in Section 3.9. In the proof of this theorem the fact that the variable conditions defined below are fulfilled for a certain recursive type will be crucial for showing the admissibility of a number of occurring substitution expressions.

Definition 3.3.15 (The variable conditions VC_0, DB and VC). Let $\tau \in \mu\mathit{Tp}$. We say that τ *fulfills the variable condition* VC_0 (denoted symbolically by $\mathit{VC}_0(\tau)$) if and only if

$$\mathrm{fv}(\tau) \cap \mathrm{bv}(\tau) = \emptyset \qquad (3.11)$$

holds, i.e. if no type variable has both a free and a bound occurrence in τ. And we say that τ *fulfills the variable condition DB* (formally abbreviated by $\mathit{DB}(\tau)$) if and only if

$$\neg\,(\exists\, p_1, p_2 \in \mathit{Pos}(\tau))\,(\exists\,\alpha \in \mathit{TVar})\,(\exists\,\tau_1, \tau_2 \in \mu\mathit{Tp}) \\ \big[\, p_1 \neq p_2 \;\;\&\;\; \tau|_{p_1} = \mu\alpha.\,\tau_1 \;\;\&\;\; \tau|_{p_2} = \mu\alpha.\,\tau_2 \,\big] \qquad (3.12)$$

holds, i.e. if τ is *distinctly bound*, that is, if all μ-bindings in τ bind different type variables. And finally, we stipulate that τ *fulfills the variable condition VC* (denoted symbolically by $\mathit{VC}(\tau)$) if and only if

$$\mathit{VC}_0(\tau) \;\&\; \mathit{DB}(\tau) \qquad (3.13)$$

holds, i.e. if no type variable occurs both bound and closed in τ and if all μ-bindings in τ bind different type variables. ⊠

Remark 3.3.16. The variable condition VC_0 corresponds with what in the literature on λ-calculus is known as the "Variable Convention" (or "Barendregt's Variable Convention"); for instance, compare [Ba81, p.26]:

> "2.1.13 VARIABLE CONVENTION. If $M_1, \ldots, M_n$ occur in a certain mathematical context (e.g. definition, proof), then in these terms all bound variables are chosen to be different from the free variables."

On the other hand, the variable condition *VC* corresponds to a property of terms in λ-calculus that in the more recent paper [VB01], which is concerned with the role of α-conversion in confluence-proofs for λ-calculus, is called "Barendregt Conventional Form" (BCF). The analogous property in λ-calculus to the property *DB* for recursive types, which demands that distinct μ-binders bind distinct variables, is called *UB* in [VB01] (variables are "*uniquely bound*" by λ-binders).

3.4 Variant Relation

As already mentioned in Section 3.1, we do not identify from the outset such recursive types that are *variants* of each other in the sense that they differ only by names of bound type variables. Neither do we introduce equivalence classes of recursive types with respect to renaming of bound type variables; our basic formal objects are just the recursive types as defined by Definition 3.1.1.

The reason for this approach is that for studying proof-theoretic interrelations between different proof systems for a notion of equality on recursive types it is, at some places, necessary to justify such transformation steps in which a subtle interplay takes place between the operations of bound-variable renaming, of substitution, and of "folding/unfolding" (see (3.35) and (3.34) in Section 3.7 below) on recursive types. In particular this is the case for the proof-theoretic justifications that we will give for the admissibility of substitution rules in known proof systems for recursive types. There, for a number of arguments a detailed analysis of the interaction between the mentioned three operations on recursive types could not be avoided.

However, the decision not to identify all variants of a recursive type 'on a syntactic level' creates the need for some explicit notation for expressing and proving precise statements involving the property of two types being variants of each other. Such a notation is provided by the following definition.

Definition 3.4.1 (Renaming of bound variables in recursive types).

(i) The relation $\rightarrow_{\text{ren}}$ between two recursive types τ_1 and τ_2, which describes an atomic admissible $\underline{\text{ren}}$aming-step of a bound variable in τ_1, is defined for all $\tau_1, \tau_2 \in \mu Tp$ by

$\tau_1 \rightarrow_{\text{ren}} \tau_2 \iff_{\text{def}}$ There exists a subterm occurrence of $\mu\alpha.\,\rho$ in τ_1 such that, for some $\tilde{\alpha}$ with $\tilde{\alpha} \not\equiv \alpha$, $\tilde{\alpha} \notin \text{fv}(\rho)$ and $\tilde{\alpha}$ substitutible for α in ρ, the recursive type τ_2 is the result of replacing this subterm occurrence of $\mu\alpha.\,\rho$ in τ_1 by $\mu\tilde{\alpha}.\,\rho[\tilde{\alpha}/\alpha]$.

(For an alternative definition of $\rightarrow_{\text{ren}}$ that relies on contexts for recursive types, see Definition 3.7.7.)

(ii) The variant relation $\equiv_{\text{ren}} \subseteq \mu Tp \times \mu Tp$ between recursive types is defined as the reflexive, transitive, and symmetrical closure $\twoheadleftarrow\!\!\twoheadrightarrow_{\text{ren}}$ of $\rightarrow_{\text{ren}}$, that is, we set $\equiv_{\text{ren}} =_{\text{def}} \twoheadleftarrow\!\!\twoheadrightarrow_{\text{ren}}$. For all $\tau_1, \tau_2 \in \mu Tp$, we call τ_1 a *variant* of τ_2 if and only if $\tau_1 \equiv_{\text{ren}} \tau_2$.

Whenever this is possible, we will adhere to the following practice of denoting variants of recursive types by using single accents: if a recursive type is denoted by a syntactical variable like $\tau, \sigma, \ldots$, then we will let variants of this recursive type be denoted by the respective one of the syntactical variables $\tau', \sigma', \ldots$.

⊠

It is rather obvious to see that $\rightarrow_{\text{ren}}$ is symmetrical, and that therefore also the transitive and reflexive closure $\twoheadrightarrow_{\text{ren}}$ of $\rightarrow_{\text{ren}}$ is symmetrical. It follows that the relations $\twoheadrightarrow_{\text{ren}}$ and $\equiv_{\text{ren}}$ are identical.

As a consequence of the decision to deal with renamings of bound variables in recursive types explicitly here, we will have to rely, at numerous places, on some basic properties of the equivalence relation $\equiv_{\text{ren}}$ between recursive types. The most frequently required properties, which will sometimes be used without explicit reference, are listed in the following lemma.

Lemma 3.4.2. *The following statements hold for all $\alpha, \beta \in TVar$, $C \in \mu Tp\text{–}Ctxt$, and for all $\tau, \tau_0, \tau_1, \tau_2, \tau', \tau_1', \tau_2', \sigma, \sigma' \in \mu Tp$:*

$$\tau \equiv_{ren} \tau' \implies fv(\tau) = fv(\tau') \tag{3.14}$$

$$\tau \equiv_{ren} \tau' \implies \tau \to \sigma \equiv_{ren} \tau' \to \sigma \;\;\&\;\; \sigma \to \tau \equiv_{ren} \sigma \to \tau' \tag{3.15}$$

$$\tau \equiv_{ren} \tau' \implies \mu\alpha.\, \tau \equiv_{ren} \mu\alpha.\, \tau' \tag{3.16}$$

$$\tau \equiv_{ren} \tau' \implies C[\tau] \equiv_{ren} C[\tau'] \tag{3.17}$$

$$\tau \equiv_{ren} \tau' \implies \tau[\sigma/\alpha] \equiv_{ren} \tau'[\sigma/\alpha] \tag{3.18}$$

$$\tau \equiv_{ren} \tau' \;\;\&\;\; \sigma \equiv_{ren} \sigma' \implies \tau[\sigma/\alpha] \equiv_{ren} \tau'[\sigma'/\alpha] \tag{3.19}$$

$$\tau \in \{\bot, \top, \alpha\} \;\;\&\;\; \tau' \equiv_{ren} \tau \implies \tau' \equiv \tau \tag{3.20}$$

$$\left.\begin{array}{l} \tau' \equiv_{ren} \tau_1 \to \tau_2 \implies (\exists\, \tau_1', \tau_2' \in \mu Tp) \\ \qquad\qquad \big[\, \tau' \equiv \tau_1' \to \tau_2' \;\;\&\;\; \tau_1' \equiv_{ren} \tau_1 \;\;\&\;\; \tau_2' \equiv_{ren} \tau_2 \,\big] \end{array}\right\} \tag{3.21}$$

$$\tau' \equiv_{ren} \mu\alpha.\, \tau_0 \implies (\exists\, \tilde{\tau}_0 \in \mu Tp)\, (\exists\, \tilde{\alpha} \in TVar)\, \big[\, \tau' \equiv \mu\tilde{\alpha}.\, \tilde{\tau}_0 \,\big] \tag{3.22}$$

$$\mu\alpha.\, \tau \equiv_{ren} \mu\alpha.\, \tau' \implies \tau \equiv_{ren} \tau' \tag{3.23}$$

$$\mu\alpha.\, \tau \equiv_{ren} \mu\beta.\, \sigma \implies \tau[\mu\alpha.\, \tau/\alpha] \equiv_{ren} \sigma[\mu\beta.\, \sigma/\beta] \tag{3.24}$$

Furthermore, also the following two statements hold, which assert that substitutions in recursive types are always facilitated by going over to appropriate variants: firstly

$$\begin{array}{l} (\forall \sigma \in \mu Tp)\, (\forall \alpha \in TVar) \\ \qquad (\forall \tau \in \mu Tp)\, (\exists\, \tau' \in \mu Tp)\, \big[\, \tau' \equiv_{ren} \tau \;\;\&\;\; \sigma \textit{ is substitutible for } \alpha \textit{ in } \tau' \,\big] \end{array} \tag{3.25}$$

holds, and as a refinement of a special case of this statement, secondly it is the case that:

$$\begin{array}{l} (\forall \sigma \in \mu Tp)\, (\forall \alpha \in TVar) \\ \qquad (\forall \mu\beta.\, \tau \in \mu Tp)\, (\exists\, \tau' \in \mu Tp)\, (\exists\, \tilde{\beta} \in TVar) \\ \qquad\qquad \big[\, \tau' \equiv_{ren} \tau \;\;\&\;\; \mu\tilde{\beta}.\, \tau'[\tilde{\beta}/\beta] \equiv_{ren} \mu\beta.\, \tau \;\;\& \\ \qquad\qquad\qquad \&\;\; \sigma \textit{ is substitutible for } \alpha \textit{ in } \mu\tilde{\beta}.\, \tau'[\tilde{\beta}/\beta] \,\big]\;. \end{array} \tag{3.26}$$

About the Proof. Some of the assertions of the lemma associated with (3.14)–(3.26) are fairly straightforward to prove; others need more subtle arguments. In particular, (3.14)–(3.17) and (3.20)–(3.22) can be shown easily by induction on the length of a conversion with respect to the reduction relation $\rightarrow_{\text{ren}}$ between τ and τ' (however, (3.16) is an immediate consequence of (3.17)). As an example of an assertion that needs a more involved proof, the statement that (3.18) holds for all $\tau, \tau', \sigma \in \mu Tp$ and $\alpha \in TVar$ is shown in Appendix A, Section A.2, starting on page 336. While not being substantially more difficult to prove, we do not give proofs here for the statements associated with (3.19) and (3.23)–(3.26).

3.5 Tree Unfolding and Leading Symbol

In this section we give formal definitions of the notions of "tree unfolding" of a recursive type and of the simpler, but related notion of "leading symbol" of a recursive type.

Definition 3.5.1 (Type Trees). A *type tree* t is a partial function

$$t : \{1,2\}^* \rightharpoonup \{\bot, \top, \to\} \cup TVar$$

with domain $Acc(t) =_{\text{def}} \text{dom}(t)$, whose members are called *access paths of* t such that the following two properties hold:

(i) $Acc(t) \neq \emptyset$ and $Acc(t)$ is prefix-closed, i.e. it holds:

$$\big(\forall p, p_1, p_2 \in \{1,2\}^*\big) \left[p \equiv p_1 p_2 \;\;\&\;\; p \in Acc(t) \;\Rightarrow\; p_1 \in Acc(t) \right] ,$$

i.e. all prefixes of access paths of t are again access paths of t.

(ii) For the arity of the symbol labeling some node p in t (which arity is 2 for the type implication symbol $\to$ and 0 for the symbols $\bot, \top$ and all type variables) is equal to the number of successors of p in t, i.e.

$$\begin{aligned}(\forall p \in \{1,2\}^*) \Big[\big(t(p) \equiv \to \;\;\Rightarrow\;\; p1, p2 \in Acc(t) \big) \;\;\& \\ \&\;\; \big(t(p) \in \{\bot, \top\} \cup TVar \;\Rightarrow\; p1, p2 \notin Acc(t) \big) \Big] .\end{aligned}$$

We denote by $TpTrees$ the *set of all type trees*.

Following the stipulations in Chapter 2, Subsection 2.1.2, the symbols $\downarrow$ and $\uparrow$ are used to designate definedness and undefinedness of a partial function on an argument, respectively. Here in particular, for a given type tree t and a path $p \in \{1,2\}^*$, the expression $t(p)\downarrow$ means that $p \in Acc(t)$, and the expression $t(p)\uparrow$ means that $p \notin Acc(t)$.

We furthermore assume the following partial order on $\{1,2\}^*$, which is induced by the "prefix"-relation and which is able to partially order the set $Acc(t)$ of access paths of an arbitrary type tree t: for all $p_1, p_2 \in \{1,2\}^*$, we let

$$\begin{aligned} p_1 \leq p_2 \quad &\Longleftrightarrow_{\text{def}} \;\; p_1 \text{ is a prefix of } p_2 \\ &\big(\Longleftrightarrow \;\; (\exists \tilde{p} \in \{1,2\}^*) \, (p_2 = p_1 \tilde{p}) \big) . \end{aligned}$$

⊠

Next we define the notion of subtree of a type tree. The well-definedness of subtrees of type trees as type trees will be stated by the subsequent proposition.

Definition 3.5.2 (Subtrees of type trees). Let $t \in \mathit{TpTrees}$ be a type tree and let $p \in \mathit{Acc}(t)$ be arbitrary. The *subtree* $t|_p$ of the type tree t *determined by* p is the partial function defined by

$$\begin{aligned} t|_p \,:\, \{1,2\}^* \;&\rightharpoonup\; \{\bot, \top, \to\} \cup \mathit{TVar} \\ \tilde{p} \;&\longmapsto\; t|_p(\tilde{p}) =_{\mathrm{def}} t(p.\tilde{p})\;. \end{aligned} \tag{3.27}$$

☒

Proposition 3.5.3. *Let* $t \in \mathit{TpTrees}$ *and* $p \in \mathit{Acc}(t)$. *Then the subtree* $t|_p$ *of* t *determined by* p *is a type tree with the property*

$$\mathit{Acc}(t|_p) = \{\tilde{p} \in \{0,1\}^* \mid p.\tilde{p} \in \mathit{Acc}(t)\}\;. \tag{3.28}$$

Proof. The representation (3.28) of the set of access paths of a subtree $t|_p$ determined by p is an obvious consequence of the definition of t_p. □

For the definition of the "tree unfolding" of a recursive type, we will need the notion of the "number of leading μ-bindings" of a recursive type.

Definition 3.5.4 (Number of leading μ-bindings of a recursive type). The *number of leading μ-bindings* $\mathrm{nl}\mu\mathrm{b}(\tau)$ of a recursive type τ is a natural number or zero that is defined by induction on the formal structure of τ (as defined in the grammar (3.1) in Definition 3.1.1) using the following clauses:

$$\mathrm{nl}\mu\mathrm{b}(\bot) =_{\mathrm{def}} 0\,, \quad \mathrm{nl}\mu\mathrm{b}(\top) =_{\mathrm{def}} 0\,, \quad \mathrm{nl}\mu\mathrm{b}(\alpha) =_{\mathrm{def}} 0 \;\; (\text{for all } \alpha \in \mathit{TVar})\,,$$
$$\mathrm{nl}\mu\mathrm{b}(\tau_1 \to \tau_2) =_{\mathrm{def}} 0 \;\; (\text{for all } \tau_1, \tau_2 \in \mu\mathit{Tp})\,,$$
$$\mathrm{nl}\mu\mathrm{b}(\mu\alpha.\,\tau_0) =_{\mathrm{def}} 1 + \mathrm{nl}\mu\mathrm{b}(\tau_0) \;\; (\text{for all } \alpha \in \mathit{TVar} \text{ and } \tau_0 \in \mu\mathit{Tp})\,.$$

☒

Example 3.5.5. It is obvious to verify that, according to the definition above,

$$\mathrm{nl}\mu\mathrm{b}(\mu\alpha.\,\alpha) = \mathrm{nl}\mu\mathrm{b}(\mu\alpha.\,\beta) = 1\,, \;\; \mathrm{nl}\mu\mathrm{b}(\alpha) = 0\,, \;\; \mathrm{nl}\mu\mathrm{b}(\mu\alpha_1 \ldots \alpha_n.\,(\rho_1 \to \rho_2)) = n$$

holds, for all $\alpha, \beta, \alpha_1, \ldots, \alpha_n \in \mathit{TVar}$ and $\rho_1, \rho_2 \in \mu\mathit{Tp}$. ■

As a first, and quite obvious, property of the function $\mathrm{nl}\mu\mathrm{b}(\cdot)$, we note that it is invariant under renaming of bound variables in recursive types.

Proposition 3.5.6. *For all* $\tau, \tau' \in \mu\mathit{Tp}$ *with* $\tau' \equiv_{\mathit{ren}} \tau$, $\mathit{nl}\mu\mathit{b}(\tau') = \mathit{nl}\mu\mathit{b}(\tau)$ *holds.*

Proof. The proposition follows from the following assertion by induction on the length of a conversion with respect to the reduction relation $\to_{\mathrm{ren}}$ between recursive types τ and τ': For all $\tau, \tau' \in \mu\mathit{Tp}$ such that $\tau \to_{\mathrm{ren}} \tau'$ is the case, it holds that $\mathrm{nl}\mu\mathrm{b}(\tau) = \mathrm{nl}\mu\mathrm{b}(\tau')$. This, in its turn, can be shown by induction on $|\tau|$, where in the induction step the following assertion is used: for all $\tau \in \mu\mathit{Tp}$ and $\alpha, \tilde{\alpha} \in \mathit{TVar}$, it holds that $\mathrm{nl}\mu\mathrm{b}(\tau[\tilde{\alpha}/\alpha]) = \mathrm{nl}\mu\mathrm{b}(\tau)$ (which can again be proved by induction on $|\tau|$). □

In the following lemma we give a characterization of the effect that "outermost-unfolding" has in a recursive type τ on the measure $\mathrm{nl}\mu\mathrm{b}(\tau)$. By "outermost-unfolding" we hereby mean the operation that takes a recursive type $\mu\alpha.\,\tau$ over to a recursive type $\tau'[\mu\alpha.\,\tau/\alpha]$, where τ' is a variant of τ with the property that $\mu\alpha.\,\tau$ is substitutible for α in τ' (for a definition of "outermost-unfolding" as a reduction relation on μTp see Definition 3.9.1).

Lemma 3.5.7. *For all $\tau, \tau' \in \mu Tp$ and $\alpha_1 \in TVar$ such that $\tau' \equiv_{ren} \tau$ and $\mu\alpha_1.\,\tau$ is substitutible for α_1 in τ' the following two (independent) equivalences hold:*

$$\left.\begin{aligned} & nl\mu b(\,\tau'[\mu\alpha_1.\,\tau/\alpha_1]\,) < nl\mu b(\mu\alpha_1.\,\tau) \quad \Longleftrightarrow \\ \Longleftrightarrow \quad & \neg\,(\exists\, n \in \omega\backslash\{0\})\,(\exists\, \alpha_2, \ldots, \alpha_n \in TVar) \\ & \qquad \big[\,\tau \equiv \mu\alpha_2 \ldots \alpha_n.\,\alpha_1 \;\;\&\;\; \alpha_1 \not\equiv \alpha_2, \ldots, \alpha_n\,\big] \\ \Longleftrightarrow \quad & nl\mu b(\,\tau'[\mu\alpha_1.\,\tau/\alpha_1]\,) = nl\mu b(\mu\alpha_1.\,\tau) - 1\;. \end{aligned}\right\} \tag{3.29}$$

A *proof* of this lemma is given in Appendix A, Section A.3, on page 338. We are now able to give the definition of the notion of "tree unfolding" of a recursive type.

Definition 3.5.8 (Tree unfolding of a recursive type). The function

$$\begin{aligned} \mathbf{Tree} : \mu Tp & \longrightarrow TpTrees \\ \tau & \longmapsto \mathbf{Tree}(\tau) \;:\; \{1,2\}^* \longrightarrow \{\bot, \top, \to\} \cup TVar \\ & \qquad\qquad\qquad\quad p \quad \longmapsto \quad \mathbf{Tree}(\tau)(p) \end{aligned}$$

that assigns to every recursive type $\tau \in \mu Tp$ its *tree unfolding* $\mathbf{Tree}(\tau)$ is defined as follows. For all $\tau \in \mu Tp$ and all $p \in \{1,2\}^*$, the symbol $\mathbf{Tree}(\tau)(p)$ is defined by induction on $|p|$ together with a sub-induction on $\mathrm{nl}\mu\mathrm{b}(\tau)$, the number of leading μ-bindings in τ, according to the five clauses below. According to the stipulations in Section 2.1.2 of Chapter 2, the symbol $\uparrow$ is used here in expressions like $\mathbf{Tree}(\tau)(p)\uparrow$ to denote undefinedness of the partial function $\mathbf{Tree}(\tau)$ on an argument $p \in \{1,2\}^*$.

(1) If $\mathrm{nl}\mu\mathrm{b}(\tau) = 0$, and $\tau \equiv \bot$ or $\tau \equiv \top$ or, for some $\alpha \in TVar$, $\tau \equiv \alpha$, then

$$\mathbf{Tree}(\tau)(\epsilon) =_{\mathrm{def}} \tau\;, \qquad\qquad \mathbf{Tree}(\tau)(p)\uparrow \quad (\text{if } p \neq \epsilon)\;.$$

(2) If $\mathrm{nl}\mu\mathrm{b}(\tau) = 0$ and $\tau \equiv \tau_1 \to \tau_2$ and $p = i\,p_0$, for some $\tau_1, \tau_2 \in \mu Tp$, $i \in \{1,2\}$ and $p_0 \in \{1,2\}^*$, then

$$\mathbf{Tree}(\tau)(\epsilon) =_{\mathrm{def}} \to\;, \qquad\qquad \mathbf{Tree}(\tau)(p) =_{\mathrm{def}} \mathbf{Tree}(\tau_{i+1})(p_0)\;.$$

(3) If $\mathrm{nl}\mu\mathrm{b}(\tau) = n$, for some $n \geq 1$, and $\tau \equiv \mu\alpha_1.\,\tau_0$, for some $\tau_0 \in \mu Tp$ and $\alpha_1 \in TVar$ with $\alpha_1 \notin \mathrm{fv}(\tau_0)$, then

$$\mathbf{Tree}(\tau)(p) =_{\mathrm{def}} \mathbf{Tree}(\tau_0)(p)\;.$$

(4) If $\mathrm{nl}\mu\mathrm{b}(\tau) = n$, for some $n \geq 1$, and $\tau \equiv \mu\alpha_1.\,\tau_0$, for some $\tau_0 \in \mu Tp$ and $\alpha_1 \in TVar$ such that with $\alpha_1 \in \mathrm{fv}(\tau_0)$, and if furthermore $\tau \equiv \mu\alpha_1 \ldots \alpha_n.\,\alpha_1$ for some type variables $\alpha_2, \ldots, \alpha_n \in TVar$ (it follows that $\alpha_1 \not\equiv \alpha_2, \ldots, \alpha_n$), then

$$\mathbf{Tree}(\tau)(\epsilon) =_{\mathrm{def}} \bot\,, \qquad\qquad \mathbf{Tree}(\tau)(p)\uparrow \quad (\text{if } p \neq \epsilon)\;.$$

(5) If $\mathrm{nl}\mu\mathrm{b}(\tau) = n$, for some $n \geq 1$, and $\tau \equiv \mu\alpha_1.\,\tau_0$, for some $\alpha_1 \in TVar$ and $\tau_0 \in \mu Tp$ such that $\alpha_1 \in \mathrm{fv}(\tau_0)$, and if furthermore $\tau \not\equiv \mu\alpha_1 \ldots \alpha_n.\,\alpha_1$ for all $\alpha_2, \ldots, \alpha_n \in TVar$ (it follows that τ must be of the form $\mu\alpha_1 \ldots \alpha_n.\,(\rho_1 \to \rho_2)$ for some $\alpha_2 \ldots \alpha_n \in TVar$ and $\rho_1, \rho_2 \in \mu Tp$), then we set:

$$\mathbf{Tree}(\tau)(p) =_{\mathrm{def}} \mathbf{Tree}(\,\tau_0'\,[\tau/\alpha_1]\,)(p)\;,$$

where τ_0' is a variant of τ_0 with the property that τ is substitutible for α in τ_0'; to disambiguate this definition (with respect to the possibility of choosing different appropriate variants τ_0' of τ_0), we furthermore assume at this point that τ_0' is actually the outcome of a computation of an effective deterministic algorithm A on input τ, where A is supposed to satisfy

> "For all $\sigma_0 \in \mu Tp$ and all variables β on input $\mu\beta.\,\sigma_0$ the algorithm A produces after a finite number of steps as output a variant σ_0' of σ_0, which has the property that $\mu\beta.\,\sigma_0$ is substitutible for β in σ_0'." (3.30)

and where A is furthermore assumed to be given as an underlying and independent tool for the entire definition of **Tree** here.[4]

☒

Some well-definedness issues in connection with this definition are considered and treated in the following remark.

Remark 3.5.9. (i) The well-definedness of the tree unfolding $\mathbf{Tree}(\tau)$ of a recursive type τ in clause (5) of Definition 3.5.8, i.e. that $\mathrm{nl}\mu\mathrm{b}(\tau_0'[\mu\alpha_1.\,\tau_0/\alpha_1]) < \; < \mathrm{nl}\mu\mathrm{b}(\mu\alpha_1.\,\tau_0)$ holds (we have $\tau \equiv \mu\alpha_1.\,\tau_0$ in this case), follows directly from Lemma 3.5.7.

(ii) It is possible to subsume the treatment of case (3) in the inductive definition of $\mathbf{Tree}(\tau)$ for $\tau \in \mu Tp$ in Definition 3.5.8 under the treatment of case (5). We have chosen not to do so for the sake of clarity and to ease the proof of a lemma below.

[4] It is clear, that algorithms with the property (3.30) can in fact be built: for example, an algorithm can be constructed, which for input $\mu\beta.\,\sigma_0$ always performs a finite sequence of $\to_{\mathrm{ren}}$-steps to σ_0 such that for the last type σ_0' of the sequence the sets $\mathrm{fv}(\sigma_0')$ of free and $\mathrm{bv}(\sigma_0')$ of bound variables of σ_0' are ultimately disjoint; for such a recursive type σ_0' it then clearly holds that $\mu\beta.\,\sigma_0$ is substitutible for β in σ_0'.

(iii) It is not hard to prove that the particular way of choosing, in item (5) of Definition 3.5.8, an appropriate variant τ_0' of τ_0 (by means of an algorithm that is not specified there but assumed to be given and underlying the entire definition) is actually of *no* consequence to the definition of the tree unfolding function **Tree**. More precisely, it holds that, if the definition of a function $\mathbf{Tree}_1 : \mu Tp \to TpTrees$ were based on the same clauses (1)–(5) as **Tree** in Definition 3.5.8, but on a different and perhaps even non-deterministic algorithm A_1 with the property (3.30), then the newly defined function $\mathbf{Tree}_1$ and the function **Tree** from the above definition would nevertheless coincide. This can be seen by a proof, which uses item (i) of the following lemma in an induction of the same form as used in Definition 3.5.8.

The following lemma states that the operation of taking the tree unfolding of a recursive type is invariant under the reduction relations $\twoheadrightarrow_{\text{ren}}$ and $\to_{\text{unfold}}$.

Lemma 3.5.10. *(i) Recursive types that are each others variant possess the same tree unfolding, i.e. for all $\tau, \tau' \in \mu Tp$ it holds that:*

$$\tau \equiv_{ren} \tau' \quad\Longrightarrow\quad \mathbf{Tree}(\tau) = \mathbf{Tree}(\tau')\ .$$

(ii) The tree unfolding of a recursive type is invariant under the operation of unfolding. This means that, for all $\alpha \in TVar$ and all $\tau_0, \tau_0' \in \mu Tp$, it holds:

$$\begin{aligned} \tau_0' \equiv_{ren} \tau_0 \ \ \&\ \ \mu\alpha.\,\tau_0 \text{ is substitutible for } \alpha \text{ in } \tau_0' \quad &\Longrightarrow \\ \Longrightarrow\quad \mathbf{Tree}(\mu\alpha.\,\tau_0) = \mathbf{Tree}(\tau_0'\,[\mu\alpha.\,\tau_0/\alpha]\,)\ .& \end{aligned}$$

A *proof* of this lemma is given in Appendix A, Section A.3, on page 339. For stating an important fact about the tree unfolding of recursive types, we need the following terminology: A type tree is called *regular* if it has only a finite number of different subtrees.

Fact 3.5.11. *Let $\sigma \in \mu Tp$. The tree unfolding $\mathbf{Tree}(\sigma)$ of a recursive type σ is a regular tree. The number of different subtrees of $\mathbf{Tree}(\sigma)$ is bounded by the size $|\sigma|$ of σ.*

This fact will be *proved* on page 63 in Section 3.9 as a corollary, Corollary 3.9.13, to a theorem about the notion defined there of “generated subterm” of a recursive type.

Next we give the definition of what we will mean by the “leading symbol of a recursive type”. We introduce two leading symbol functions, one that is a partial and another one that is total, and we will notice shortly that the second is an extension of the first.

Definition 3.5.12 (Leading symbol functions $\mathcal{L}$ and $\mathcal{L}'$). We define the functions $\mathcal{L}$, $\mathcal{L}'$ and the leading symbol of a recursive type in the following three items:

(i) The leading symbol function $\mathcal{L} : \ \mu Tp \rightharpoonup \{\bot, \top, \to\} \cup TVar$ is a partial function that is defined on the subset *can*-μTp of μTp by the clauses

$$\begin{gathered}
\mathcal{L}(\bot) =_{\text{def}} \bot \,, \qquad \mathcal{L}(\top) =_{\text{def}} \top \,, \qquad \mathcal{L}(\alpha) =_{\text{def}} \alpha \quad (\text{for all } \alpha \in TVar) \,, \\
\mathcal{L}(\tau_1 \to \tau_2) =_{\text{def}} \to \quad (\text{for all } \tau_1, \tau_2 \in TVar) \,, \\
\mathcal{L}(\mu\alpha.\,(\tau_1 \to \tau_2)) =_{\text{def}} \qquad (\text{for all } \tau_1, \tau_2 \in TVar \text{ and } \alpha \in TVar) \,,
\end{gathered}$$

and which is undefined on $\mu Tp \setminus$ *can*-μTp (i.e. $\mathcal{L}$ is defined for precisely all recursive types in canonical form).

(ii) The leading symbol function $\mathcal{L}' : \mu Tp \longrightarrow \{\bot, \top, \to\} \cup TVar$ is defined for all recursive types $\tau \in \mu Tp$ by

$$\mathcal{L}'(\tau) =_{\text{def}} \mathbf{Tree}(\tau)(\epsilon) \,, \tag{3.31}$$

i.e. $\mathcal{L}'(\tau)$ is that symbol, which labels the root in the tree unfolding $\mathbf{Tree}(\tau)$ of τ.

(iii) For every recursive type $\tau \in \mu Tp$, the *leading symbol* of τ is defined as $\mathcal{L}'(\tau)$. $\boxtimes$

Remark 3.5.13. As we will see in the following proposition, the leading-symbol function $\mathcal{L}'$ is the extension of the partial leading-symbol function $\mathcal{L}$ to the set μTp of all recursive types. We have defined the partial function $\mathcal{L}$ separately from its proper extension $\mathcal{L}'$ here in an effort to make the slightly more involved definition of $\mathcal{L}'$ directly accessible in the special case of defining the leading symbol of a recursive type in canonical form.

It is easy to see that the definition of $\mathcal{L}'$ can be given in a more explicit and non-inductive way by expanding the right side of (3.31) according to the definition of the tree unfolding in Definition 3.5.8. We have avoided doing this here because we would essentially have repeated a special case of the clauses (i)–(iv) in Definition 3.5.8. However, it is important to note that the definition of the leading symbol $\mathcal{L}'(\tau)$ of a recursive type τ does not presuppose full knowledge of the tree unfolding $\mathbf{Tree}(\tau)$ of τ, which usually is infinite. In fact, $\mathcal{L}'(\tau)$ can actually always be produced from τ 'in a linear amount of time'.

Proposition 3.5.14. *The leading symbol function $\mathcal{L}'$ is an extension of the leading symbol function $\mathcal{L}$ to a total function with the set μTp of recursive types as its domain. In particular, $\mathcal{L}'(\tau) = \mathcal{L}(\tau)$ holds for all $\tau \in$ can-μTp.*

Proof. Due to the definition of $\mathcal{L}'$ via the notion of tree unfolding, it suffices to show that the leading symbol functions $\mathcal{L}$ and $\mathcal{L}'$ agree on *can*-μTp, the set of recursive types in canonical form.

The statement, that, for all $\tau \in$ *can*-μTp, $\mathcal{L}'(\tau) = \mathcal{L}(\tau)$ holds, can be shown by distinguishing the five possible cases for the last generation step of τ with respect to the defining grammar (3.2) for the set *can*-μTp. Here, we consider only the not entirely obvious case that $\tau \equiv \mu\alpha.\,(\tau_1 \to \tau_2)$ for some $\tau, \tau_1, \tau_2 \in \mu Tp$ and $\alpha \in TVar$. In this case we find, by expanding the definitions of $\mathcal{L}'$, of **Tree**, and of $\mathcal{L}$, that, with appropriate respective variants τ_1' and τ_2' of τ_1 and τ_2, the following holds:

$$\begin{aligned}\mathcal{L}'(\mu\alpha.\,(\tau_1 \to \tau_2)) &= \mathbf{Tree}(\mu\alpha.\,(\tau_1 \to \tau_2))(\epsilon)\\ &= \mathbf{Tree}(\tau_1'[\tau/\alpha] \to \tau_2'[\tau/\alpha]) \;=\; \to \;=\; \mathcal{L}(\mu\alpha.\,(\tau_1 \to \tau_2))\ .\end{aligned}$$

Hence the functions $\mathcal{L}'$ and $\mathcal{L}$ do indeed coincide on τ in this case. □

Example 3.5.15. We want to find the leading symbols of the recursive types $\mu\alpha_1\alpha_2.\,\alpha_2$, $\mu\alpha_1\alpha_2.\,\alpha_3$, and $\mu\alpha.\,(\alpha \to \bot)$. By expanding the definitions of the leading-symbol function $\mathcal{L}'$ and of the tree unfolding function **Tree**, we find

$$\mathcal{L}'(\mu\alpha_1\alpha_2.\,\alpha_2) = \mathbf{Tree}(\mu\alpha_1\alpha_2.\,\alpha_2)(\epsilon) = \mathbf{Tree}(\mu\alpha_2.\,\alpha_2)(\epsilon) = \bot\ ,$$
$$\mathcal{L}'(\mu\alpha_1\alpha_2.\,\alpha_3) = \mathbf{Tree}(\mu\alpha_1\alpha_2.\,\alpha_3)(\epsilon) = \mathbf{Tree}(\mu\alpha_2.\,\alpha_3)(\epsilon) = \mathbf{Tree}(\alpha_3)(\epsilon) = \alpha_3\ ,$$
$$\mathcal{L}'(\mu\alpha.\,(\alpha \to \bot)) = \mathbf{Tree}(\mu\alpha.\,(\alpha \to \bot))(\epsilon) = \mathbf{Tree}((\mu\alpha.\,(\alpha \to \bot)) \to \bot)(\epsilon) = \to\ .$$

Hence the leading symbols of $\mu\alpha_1\alpha_2.\,\alpha_2$, $\mu\alpha_1\alpha_2.\,\alpha_3$, and $\mu\alpha.\,(\alpha \to \bot)$ are $\bot$, α_3, and →, respectively. ■

We conclude this section with a proposition that characterizes those recursive types that have the type composition symbol → as their leading symbol.

Proposition 3.5.16. *For all $\tau \in \mu Tp$ it holds:*

$$\left.\begin{aligned}\mathcal{L}'(\tau) \;=\to \quad\Longleftrightarrow\quad &(\exists\, n \in \omega)\,(\exists\, \alpha_1, \ldots, \alpha_n \in TVar)\\ &\qquad(\exists\, \rho_1, \rho_2 \in \mu Tp)\,\big[\,\tau \equiv \mu\alpha_1 \ldots \alpha_n.\,(\rho_1 \to \rho_2)\,\big]\ .\end{aligned}\right\} \quad (3.32)$$

Proof. The proposition follows from the chain of equivalences (7.7) that are justified, and used, in the proof of Lemma 7.1.1 in Chapter 7. □

3.6 Recursive Type Equality

Based on the notion of tree unfolding of recursive types, it is now possible to define the relation "recursive type equality", or "strong equivalence", on recursive types.

Definition 3.6.1 (Recursive Type Equality (strong recursive type equivalence)). Two recursive types τ and σ are called *strongly equivalent* (which we denote symbolically by $\tau =_\mu \sigma$) if and only if they possess the same tree unfolding. And furthermore, the relation $=_\mu \subseteq \mu Tp \times \mu Tp$ that is defined by stipulating, for all $\tau, \sigma \in \mu Tp$,

$$\tau =_\mu \sigma \quad\Longleftrightarrow_{\text{def}}\quad \mathbf{Tree}(\tau) = \mathbf{Tree}(\sigma) \tag{3.33}$$

is called *recursive type equality*, or *strong recursive type equivalence* (shorter, *strong equivalence*), in accordance with already introduced symbolic notation for "strongly equivalent". ⊠

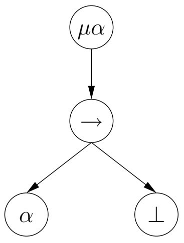
We are going to give two examples for the above defined notions of the tree unfolding of a recursive type and of the strong recursive type equivalence relation. Instead of arguing formally precise, we thereby admit a more informal argumentation, that involves pictures of labeled trees and of cyclic term graphs. In particular, we use a graphical representation of recursive types as cyclic term graphs. We thereby base ourselves on what is surely a *standard translation* of recursives types τ into cyclic term graphs $G(\tau)$ that is made precise in the more general setting of μ-terms over a first-order signature in the report version of [ArKl95]. It can be found there in Definition 2.7 on page 7 (in an informal and graphical but rather instructive way) as well as in Section 5.1 on page 36 (there in a formal way, by which a recursion system—that is itself assumed to represent a cyclic term graph uniquely—is assigned to an arbitrary given μ-term over a first order signature).

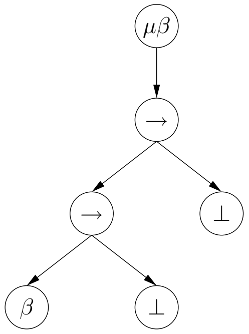
For the sake of the two examples below, we also assume tacitly a soundness statement for our informal argumentation, which asserts, for all $\tau \in \mu Tp$, that the outcome $\mathbf{Tree}(G(\tau))$ of taking the tree unfolding (in a "canonically" defined way[5]) of the cyclic-term graph $G(\tau)$ associated in the above mentioned way with τ is actually equal to $\mathbf{Tree}(\tau)$ as defined in Definition 3.5.8. – We do not prove this statement here, since (1) it is beyond the scope of the present study, and because (2) it is not inherently necessary for the purpose at hand here of constructing and convincing ourselves of the geometrical shape of the tree unfolding $\mathbf{Tree}(\tau)$ of a given recursive type τ as defined in Definition 3.5.8 (but the detour via the cyclic term graph $G(\tau)$ can provide some helpful visualization).

Example 3.6.2 (Tree unfolding, strong recursive type equivalence). We consider the recursive types in canonical form

$$\tau_1 \equiv \mu\alpha.\,(\alpha \to \bot) \qquad \text{and} \qquad \sigma_1 \equiv \mu\beta.\,((\beta \to \bot) \to \bot)\ .$$

The term trees of τ_1 and σ_1 are of respective shape

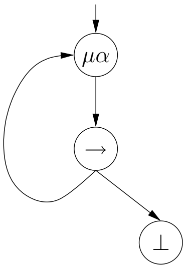


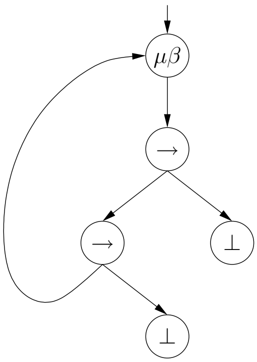


From these term trees the cyclic term graphs that correspond, under the mentioned standard translation from [ArKl95], to τ_1 and σ_1, respectively, can be found as

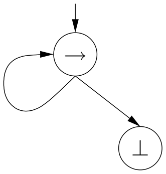
[5] Building the tree unfolding $\mathbf{Tree}(g)$ of a given cyclic term graph g means informally: to associate with every path p in the graph g, which leads to a node n in g with label l, a node n_p in the tree unfolding $\mathbf{Tree}(g)$ of g, which is accessible from the root of $\mathbf{Tree}(g)$ by a path corresponding to p, and which node n_p carries also label l.

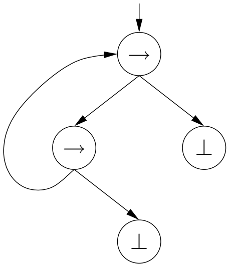
follows: firstly, remove nodes corresponding to bound variables and redirect the respective incoming edge to the node with the μ-binding by which the variable was bound, thereby reaching the cyclic graphs





And secondly, "collapse" nodes with μ-bindings to the nearest node below that is labeled by a symbol → (if, other than in the example here, such a node does not exist, "collapse" to a "black hole"), carrying backbindings to the node with the μ-binding along during this movement. In this way we find the cyclic term graphs

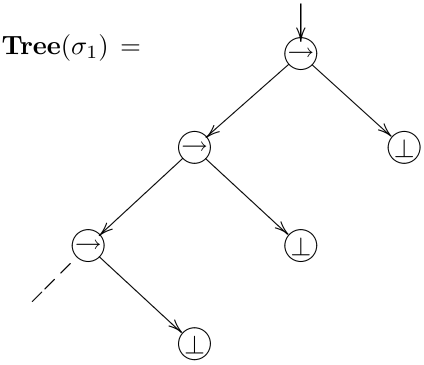




as the result here.

Now it is easy to see that these two cyclic term graphs have the same tree unfolding, see the picture of the labeled tree below. And also it can be shown by induction on the length of paths $p \in \{1, 2\}^*$) that τ_1 and τ_2 possess the same tree unfolding. However, it is easy to show (by induction on the length of paths $p \in \{1, 2\}^*$) that τ_1 and τ_2 possess the same tree unfolding, a type tree of the form

$\mathbf{Tree}(\tau_1) \;=\; \mathbf{Tree}(\sigma_1) \;=$

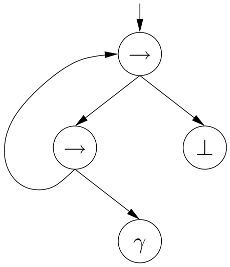


Such a (type) tree is sometimes called a “comb” because of its geometrical shape. Hence we find $\tau_1 =_\mu \sigma_1$, i.e. that τ_1 and σ_1 are strongly equivalent, by Definition 3.6.1.

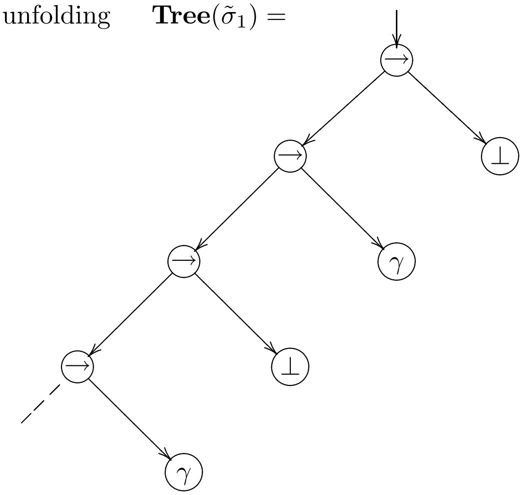
The situation is different, however, if we consider the slightly different recursive type in canonical form

$$\tilde{\sigma}_1 \equiv \mu\beta.\,((\beta \to \gamma) \to \bot) \qquad (\text{where } \gamma \not\equiv \beta)$$

in place of σ_1. The recursive type $\tilde{\sigma}_1$ corresponds, via the mentioned standard translation, to the cyclic term graph



and has the tree unfolding $\mathbf{Tree}(\tilde{\sigma}_1) =$



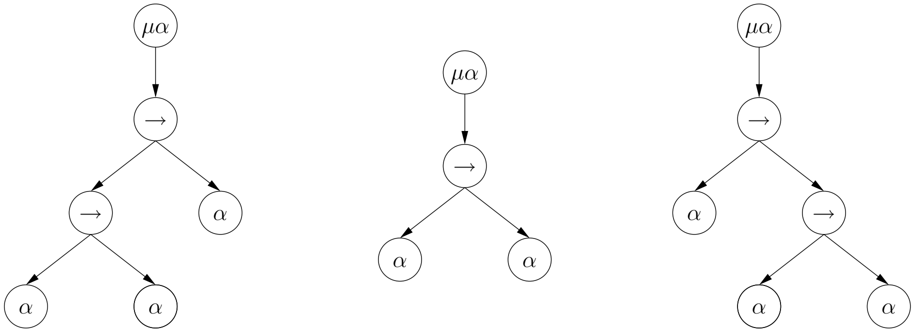
Obviously $\mathbf{Tree}(\tilde{\sigma}_1)$ is different from $\mathbf{Tree}(\tau_1)$, and hence $\tau_1 \neq_\mu \tilde{\sigma}_1$ follows. Analogously we recognize that also $\sigma_1 \neq_\mu \tilde{\sigma}_1$ is the case. Thus the recursive type $\tilde{\sigma}_1$ is *not* strongly equivalent to either of the recursive types τ_1 or σ_1. ■

We continue with giving a second example.

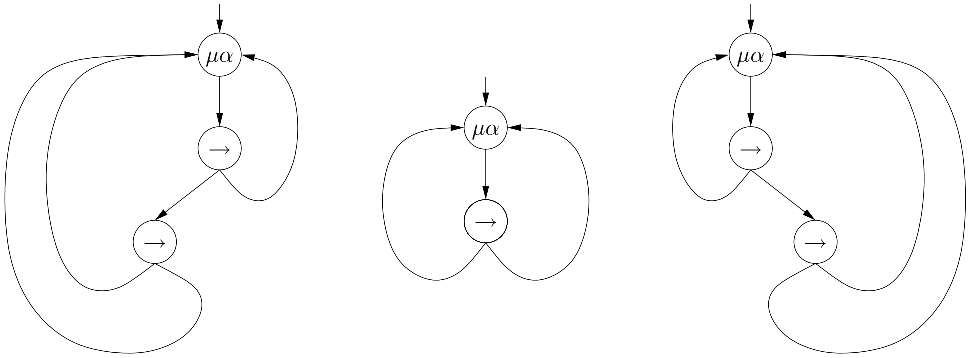
Example 3.6.3 (Tree unfolding, strong recursive type equivalence). We consider the three recursive types in canonical form

$$\tau_2 \equiv \mu\alpha.\,((\alpha \to \alpha) \to \alpha)\ , \quad \rho_2 \equiv \mu\alpha.\,(\alpha \to \alpha)\ , \quad \text{and} \quad \sigma_2 \equiv \mu\alpha.\,(\alpha \to (\alpha \to \alpha))\ .$$

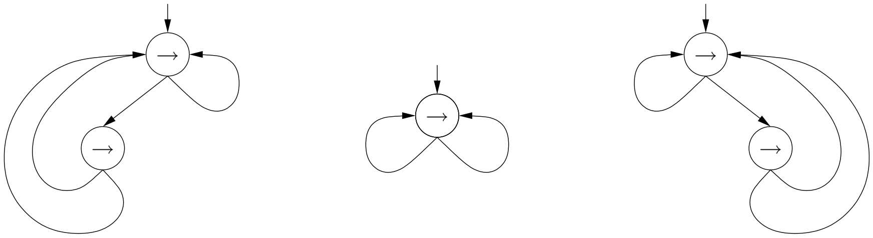
The term trees of τ_2, ρ_2 and σ_2 are of the respective shape



By redirecting arrows to bound variables at the bottom of these term trees to the respective nodes with the μ-binding by which they are bound, we arrive here at the cyclic graphs

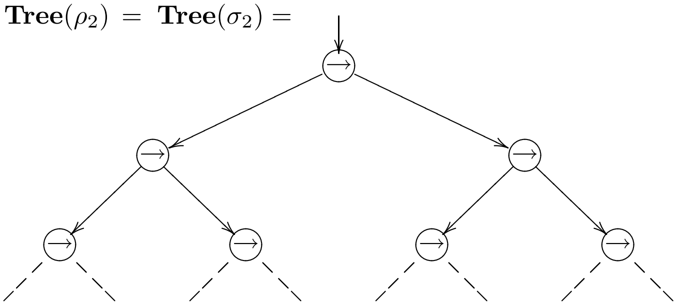


And by a second step of collapsing nodes with a μ-binding to the nearest nodes with a symbol → below, we reach the two cyclic term graphs



From these cyclic term graphs it is easy to guess that τ_2, ρ_2, and σ_2 possess actually the same tree unfolding, namely a type tree of the form

$\mathbf{Tree}(\tau_2) \ = \ \mathbf{Tree}(\rho_2) \ = \ \mathbf{Tree}(\sigma_2) =$



Also, this can easily be verified formally.

Hence by Definition 3.6.1 each two of the three recursive types τ_2, ρ_2, and σ_2 are strongly equivalent. ■

3.7 Weak Recursive Type Equivalence

In this section we introduce a weaker notion of equality between recursive types and gather some important results about it. In contrast with strong recursive type equivalence, which is defined via the semantical denotation of recursive types as type trees, the notion of "weak recursive type equivalence" is defined in a syntactical way: Two recursive types τ and σ are called "weakly equivalent" if the equation $\tau = \sigma$ is derivable from a few basic axioms that formulate some obvious properties of strong recursive type equivalence $=_\mu$. We will give a definition of this weaker notion of equality by means of a Hilbert-style proof system, present a characterization for it via a certain reduction relation, and show that it is in fact weaker than $=_\mu$.

Whereas the relation of "strong recursive type equivalence" on recursive types was suggested and used for a type assignment system with recursive types by Cardone and Coppo[6] in [CaCo91], some earlier type assignment systems with recursive types were based on a weaker notion of equivalence; in [CaCo91] references to [MPS86] and [Men86] are given in this respect. This weaker concept of recursive type equivalence is, considered on the set of recursive types in canonical form, the smallest congruence relation on *can*-μTp that is generated by the operations of *unfolding* and *folding*, which are defined by the rewrite rules

$$\mu\alpha.\,\tau \ \rightarrow_{\text{unfold}} \ \tau[\mu\alpha.\,\tau/\alpha] \ (\text{for all } \alpha \in \mathit{TVar},\ \tau \in \mu Tp) \quad \text{and} \tag{3.34}$$

$$\sigma[\mu\beta.\,\sigma/\beta] \ \rightarrow_{\text{fold}} \ \mu\beta.\,\sigma \qquad (\text{for all } \beta \in \mathit{TVar},\ \sigma \in \mu Tp) \tag{3.35}$$

on the set μTp of recursive types; for a formal definition of $\rightarrow_{\text{fold}}$ and $\rightarrow_{\text{unfold}}$ as reduction relations on μTp see Definition 3.7.7 below. We will introduce this congruence relation, which is generally called "weak (recursive type) equivalence", by means of an axiom system that has the equations between recursive types as

[6] Coppo and Cardone did not use any of the terms "strong recursive type equivalence" or "recursive type equality" in [CaCo91].

Figure 3.2: The formal system **EQL** of *equational logic* with equations between recursive types in μTp as its formulas.

The *axioms* of **EQL**:

$$\text{(REFL)}\ \overline{\tau = \tau}$$

The *inference rules* of **EQL**:

$$\frac{\sigma = \tau}{\tau = \sigma}\ \text{SYMM} \qquad \frac{\tau = \rho \qquad \rho = \sigma}{\tau = \sigma}\ \text{TRANS}$$

$$\frac{\tau = \sigma}{\tau[\rho/\alpha] = \sigma[\rho/\alpha]}\ \text{SUBST} \qquad \frac{\tau = \sigma}{C[\tau] = C[\sigma]}\ \text{CTXT}$$

its formulas and that is an extension of the basic system of "equational logic on recursive types" (which is also introduced below). And we will show that the formal system that we introduce for this purpose does not axiomatize strong recursive type equivalence $=_\mu$ *completely*. However, a proof system as the one introduced below for "weak recursive type equality" is likely to have played the role of an important stepping stone towards the formalization of a sound *and* complete axiom system for strong recursive type equivalence by Amadio and Cardelli (for this system, see Chapter 5).

For the purpose of introducing proof systems for notions of equality between recursive types, we first fix some notation for sets of "equations between recursive types".

Definition 3.7.1 (Equations between recursive types). An *equation between recursive types* is a formula of the form $\tau = \sigma$, where $\tau, \sigma \in \mu Tp$ and where the equality symbol $=$ acts as a predicate symbol. The set of all equations between recursive types is designated by μTp–Eq.

Accordingly, an *equation between recursive types in canonical form* is an expression of the form $\tau = \sigma$, where $\tau, \sigma \in$ *can*-μTp. The set of all equations of recursive types in canonical form is designated by *can*-μTp–Eq. ⊠

The basic formal system of "equational logic on recursive types" formalizes five basic properties of notions of equality on the set of recursive types, namely, reflexivity, symmetry, transitivity, closedness under substitution, and closedness under contexts. It is defined as follows.

Definition 3.7.2 (Equational logic on recursive types). The (pure) Hilbert-style proof system **EQL** of *equational logic on recursive types* has the equations between recursive types in μTp–Eq as its *formulas*. Its *axioms* are all those that belong to the scheme (REFL) shown in Figure 3.2. And as its *inference rules* **EQL** contains precisely the rules SYMM, TRANS, SUBST, and CTXT that are schematically defined in Figure 3.2. ⊠

Figure 3.3: The axiom system **WEQ** for weak recursive type equality $=_{w\mu}$.

The *axioms* of **WEQ**:

(REFL) $\overline{\tau = \tau}$ (REN) $\overline{\tau = \tau'}$ (if $\tau \equiv_{\text{ren}} \tau'$)

$(\mu - \bot)$ $\overline{\mu\alpha.\,\alpha = \bot}$ (FOLD/UNFOLD) $\overline{\mu\alpha.\,\tau = \tau[\mu\alpha.\,\tau/\alpha]}$

The *inference rules* of **WEQ**:

$$\frac{\sigma = \tau}{\tau = \sigma}\ \text{SYMM} \qquad \frac{\tau = \rho \qquad \rho = \sigma}{\tau = \sigma}\ \text{TRANS}$$

$$\frac{\tau = \sigma}{\mu\alpha.\,\tau = \mu\alpha.\,\sigma}\ \mu\text{-COMPAT} \qquad \frac{\tau_1 = \sigma_1 \qquad \tau_2 = \sigma_2}{\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2}\ \text{ARROW}$$

Now we are going to introduce the proof system **WEQ**. Later we will see that **WEQ** forms the basis for a complete axiomatization of $=_\mu$ due to Amadio and Cardelli, which will be presented in Chapter 5. Actually only one additional rule will be necessary to extend **WEQ** to a complete axiom system for $=_\mu$.

Definition 3.7.3 (The proof system WEQ). The (pure) Hilbert-style proof system **WEQ** is defined as the following formal system: The *formulas* of **WEQ** are the equations between recursive types, i.e. all elements in $\mu Tp\text{–}Eq$. The *axioms* of **WEQ** are all those equations between recursive types that belong to one of the four different schemes (REFL), (VAR), $(\mu - \bot)$ and (FOLD/UNFOLD) depicted in Figure 3.3. The *inference rules* of **WEQ** are the rules SYMM, TRANS, ARROW and μ-COMPAT whose respective applications are schematically defined in Figure 3.3. ☒

Relying on the axiom system **WEQ**, the relation of "weak recursive type equivalence" is now defined via (formula) derivability in **WEQ**.

Definition 3.7.4 (Weak recursive type equivalence). The relation $=_{w\mu} \subseteq$ $\subseteq \mu Tp \times \mu Tp$, called *weak recursive types equivalence*, is defined by stipulating, for all $\tau, \sigma \in \mu Tp$,

$$\tau =_{w\mu} \sigma \quad \Longleftrightarrow_{\text{def}} \quad \vdash_{\mathbf{WEQ}} \tau = \sigma \ . \tag{3.36}$$

If, for some $\tau, \sigma \in \mu Tp$, $\tau =_{w\mu} \sigma$ holds, then we say that τ and σ are *weakly equivalent*. We will permit ourselves to speak of $=_{w\mu}$ just as of *weak equivalence*. ☒

It is not entirely obvious that the system **WEQ** is an extension of the system **EQL** for equational logic, because the rules SUBST for substitution, and CTXT for compatibility with contexts of **EQL** are not part of the system **WEQ**. However, it turns out that both of these rules are admissible in **WEQ**, and as a consequence, that every theorem of **EQL** is also a theorem of **WEQ**.

Proposition 3.7.5. *(i) The rule* SUBST *of* **EQL** *is admissible in* **WEQ***, and the rule* CTXT *of* **EQL** *is derivable in* **WEQ***.*

(ii) **WEQ** *is an extension of* **EQL***.*

The *proof* of this proposition is deferred to Appendix A, where it is given on page 340 in Section A.4. There are two reasons for that: firstly, Proposition 3.7.5 will not be used later on, but it is given here only with the intention of signalizing basic properties of the systems **EQL** and **WEQ** and of describing their relationship towards each other; and secondly, it turns out that this proposition can be demonstrated rather easily by using special cases of results and proofs given in later chapters (the proposition is actually proven in such a way in Appendix A).

Remark 3.7.6. The axioms (REN) of **WEQ** that formulate bound variable renaming in recursive types could be replaced by the axioms

$$\text{(RENSTEP)}\quad \overline{\mu\alpha.\,\tau = \mu\beta.\,\tau[\beta/\alpha]} \qquad (\text{if } \beta \not\equiv \alpha \text{ and } \beta \notin \text{fv}(\tau))\ ,$$

which are equations $\sigma_1 = \sigma_2$ between recursive types σ_1 and σ_2 where σ_2 results from σ_1 by an atomic renaming step of bound variables that takes place at the outermost position in σ_1. More precisely, the formal system **WEQ′** that differs from **WEQ** only by the absence of the axioms (REN) and by the additional presence of the axioms (RENSTEP) is equivalent to **WEQ**. This is a consequence of the fact that the rule CTXT is also derivable in **WEQ′** (because in the proof of Proposition 3.7.5, (i), on page 340, axioms (REN) are never used for mimicking CTXT-applications by **WEQ**-derivations) and that, due to this and the presence of TRANS in **WEQ**, the axioms (REN) are derivable in **WEQ′**.

We also want to give a characterization of the relation $=_{w\mu}$ in terms of a certain rewriting relation on μTp. For this we will need the following definition.

Definition 3.7.7 (The rewrite relations $\rightarrow_{\textbf{unfold}}$, $\rightarrow_{\textbf{fold}}$, $\rightarrow_{\textbf{ren}}$, $\rightarrow_{(\textbf{unf/ren})}$, $\rightarrow_{(\mu\text{–}\bot)}$, and $\rightarrow_{(\textbf{unf/ren}/\mu\bot)}$ on the set μTp).
The rewrite relations $\rightarrow_{\text{unfold}}$, $\rightarrow_{\text{fold}}$, $\rightarrow_{\text{ren}}$, $\rightarrow_{(\mu\text{–}\bot)}$, and $\rightarrow_{(\text{unf/ren})}$ on μTp are defined as subsets of $\mu Tp \times \mu Tp$:

$$\begin{aligned}
\rightarrow_{\text{unfold}} &=_{\text{def}} \left\{ \big\langle C[\mu\alpha.\,\tau],\, C[\tau[\mu\alpha.\,\tau/\alpha]] \big\rangle \;\middle|\; \begin{array}{l} \alpha \in \mathit{TVar},\ \tau \in \mu Tp, \\ C \in \mu Tp\text{–}\mathit{Ctxt} \end{array} \right\} \\
\rightarrow_{\text{fold}} &=_{\text{def}} \leftarrow_{\text{unfold}} \\
\rightarrow_{\text{ren}} &=_{\text{def}} \left\{ \big\langle C[\mu\alpha.\,\tau],\, C[\mu\tilde{\alpha}.\,\tau[\tilde{\alpha}/\alpha]] \big\rangle \;\middle|\; \begin{array}{l} \alpha, \tilde{\alpha} \in \mathit{TVar},\ \tilde{\alpha} \not\equiv \alpha\,, \\ \tau \in \mu Tp\,,\ \tilde{\alpha} \notin \text{fv}(\tau) \end{array} \right\} \\
\rightarrow_{(\mu\text{–}\bot)} &=_{\text{def}} \left\{ \big\langle C[\mu\alpha.\,\alpha],\, C[\bot] \big\rangle \mid \alpha \in \mathit{TVar},\ C \in \mu Tp\text{–}\mathit{Ctxt} \right\} \\
\rightarrow_{(\text{unf/ren})} &=_{\text{def}} \rightarrow_{\text{unfold}} \cup \rightarrow_{\text{ren}} \\
\rightarrow_{(\text{unf/ren}/\mu\bot)} &=_{\text{def}} \rightarrow_{\text{unfold}} \cup \rightarrow_{\text{ren}} \cup \rightarrow_{(\mu\text{–}\bot)}
\end{aligned}$$

Hereby we have given for the relation $\rightarrow_{\text{ren}}$, which has already been introduced in Definition 3.4.1, an alternative, but equivalent, definition (this is easy to see). The relation $\leftarrow_{\text{unfold}}$ denotes the inverse of the relation $\rightarrow_{\text{unfold}}$, according to our stipulations in Subsection 2.1.5, Chapter 2. These stipulations are also enacted for the other reduction relations defined here: for example, $\twoheadleftarrow\!\!\twoheadrightarrow_{(\text{unf/ren}/\mu\perp)}$ denotes the convertibility relation with respect to the rewrite relation $\rightarrow_{(\text{unf/ren}/\mu\perp)}$. ⊠

With these notions it is now possible to formulate the following characterization of weak recursive type equivalence $=_{w\mu}$.

Lemma 3.7.8 (A characterization of $=_{w\mu}$). *For all $\tau, \sigma \in \mu Tp$ it holds that:*

$$\tau =_{w\mu} \sigma \quad \Longleftrightarrow \quad \tau \twoheadleftarrow\!\!\twoheadrightarrow_{(\text{unf/ren}/\mu\perp)} \sigma \ .$$

Proof. The direction "⇒" can be shown by an easy induction on the depth $|\mathcal{D}|$ of an arbitrary derivation $\mathcal{D}$ in **WEQ** with conclusion $\tau = \sigma$ (for arbitrary $\tau, \sigma \in \mu Tp$). In this induction the property is used, that the conversion $\twoheadleftarrow\!\!\twoheadrightarrow_{(\text{unf/ren}/\mu\perp)}$ is *compatible with μTp-contexts* (or: is "closed under context-formation in μTp"), i.e. that it has the property that, whenever $\tau \twoheadleftarrow\!\!\twoheadrightarrow_{(\text{unf/ren}/\mu\perp)} \sigma$ is true for some $\tau, \sigma \in \mu Tp$, then also $C[\tau] \twoheadleftarrow\!\!\twoheadrightarrow_{(\text{unf/ren}/\mu\perp)} C[\sigma]$ holds for arbitrary contexts $C \in \mu Tp\text{–}Ctxt$. The conversion $\twoheadleftarrow\!\!\twoheadrightarrow_{(\text{unf/ren}/\mu\perp)}$ does have this property, because each of the the rewrite relations $\rightarrow_{\text{unfold}}$, $\rightarrow_{\text{ren}}$ and $\rightarrow_{(\mu-\perp)}$ is actually compatible with μTp-contexts (these facts can be observed easily from the definition of these relations in Definition 3.7.7).

The direction "⇐" follows by an induction of a conversion $\twoheadleftarrow\!\!\twoheadrightarrow_{(\text{unf/ren}/\mu\perp)}$ between τ and σ (where τ and σ are arbitrary recursive types), in which induction the presence of the rules SYMM and TRANS in **WEQ** and the assertion

$$(\forall \tau, \sigma \in \mu Tp) \left[\tau \rightarrow_{(\text{unf/ren}/\mu\perp)} \sigma \quad \Longrightarrow \quad \vdash_{\mathbf{WEQ}} \tau = \sigma \right] \tag{3.37}$$

is used. (3.37) follows from the presence of the axioms (FOLD/UNFOLD), (VAR) and $(\mu - \perp)$ in **WEQ** and from the above mentioned fact that the rule CTXT of **EQL** is a derivable rule in **WEQ**. (As a consequence of the fact that the rule CTXT is a derivable rule in **WEQ**, the relation $=_{w\mu}$ of weak recursive type equality is also compatible with μTp-contexts).

□

We proceed by giving the soundness theorem for **WEQ** with respect to $=_\mu$.

Lemma 3.7.9 (Soundness of WEQ with respect to $=_\mu$). *The system* **WEQ** *is a sound axiomatization of strong recursive type equivalence. This means that for all $\tau, \sigma \in \mu Tp$ the following is true:*

$$(\, \tau =_{w\mu} \sigma \quad \Longleftrightarrow \,) \quad \vdash_{\mathbf{WEQ}} \tau = \sigma \qquad \Longrightarrow \qquad \tau =_\mu \sigma \ .$$

Put differently, this means, that $=_{w\mu} \subseteq =_\mu$ is true, i.e. that weak recursive type equivalence is contained in strong recursive type equivalence.

Although this lemma could certainly be proved directly here, we defer to give a *proof* to page 340 in Section A.4 of Appendix A; there, the lemma is settled as a special case of the soundness theorem, due to Amadio and Cardelli, for their extension $\mathbf{AC}^{=}$ of **WEQ** which will be introduced in Section 5.1 of Chapter 5.

Notwithstanding the soundness of **WEQ** with respect to $=_{\mu}$, we find that this system is actually *not complete* with respect to $=_{\mu}$.

Lemma 3.7.10. *There exist recursive types τ and σ such that τ and σ are strongly, but not weakly equivalent. This entails* $=_{w\mu} \not\supseteq =_{\mu}$.

Proof. We shall consider the recursive types

$$\begin{aligned} \tau_1 &\equiv \mu\alpha.\,(\alpha \to \bot) \qquad\qquad \text{and} \\ \sigma_1' &\equiv \mu\alpha.\,((\alpha \to \bot) \to \bot)\ , \end{aligned}$$

where we have already used τ_1 in Example 3.6.2 and where σ_1' is a variant of the recursive type $\sigma_1 \equiv \mu\beta.\,((\beta \to \bot) \to \bot)$ in Example 3.6.2. We saw in Example 3.6.2, that τ_1 and σ_1 are strongly equivalent. From this it follows with Lemma 3.5.10, (i), that then also τ_1 and σ_1' are strongly equivalent, i.e. that $\tau_1 =_{\mu} \sigma_1'$ holds.

We will show now, that τ_1 and σ_1' are not weakly equivalent. Using the characterization of $=_{w\mu}$ in Lemma 3.7.8, the assertion that for all $\rho \in \mu Tp$

$$\tau_1 =_{w\mu} \rho \implies \rho \equiv (\ldots((\tau_1' \to \underbrace{\bot) \to \bot)\ldots \to \bot}_{n})$$
$$\text{for some } n \in \omega \text{ and a variant } \tau_1' \text{ of } \tau_1$$

holds, can be proved easily by induction on the length of a conversion $\twoheadleftarrow\!\!\twoheadrightarrow_{(\text{unf/ren}/\mu\bot)}$ (which conversion must clearly also be a conversion $\twoheadleftarrow\!\!\twoheadrightarrow_{(\text{unf/ren})}$) between τ_1 and ρ. But since clearly $\tau_1' \not\equiv_{\text{ren}} \sigma_1'$ holds for all $\tau_1' \in \mu Tp$ with $\tau_1' \equiv_{\text{ren}} \tau_1$ (because actually $\tau_1 \not\equiv_{\text{ren}} \sigma_1'$ is the case), $\tau_1 =_{w\mu} \sigma_1'$ cannot be the case. Thus $\tau_1 \neq_{w\mu} \sigma_1'$ follows. □

To summarize the facts proved about the relationship between $=_{w\mu}$ and $=_{\mu}$ we formulate the following theorem:

Theorem 3.7.11. *Weak recursive type equivalence is properly contained in strong recursive type equivalence, i.e. it holds:* $=_{w\mu} \subsetneqq =_{\mu}$.

Proof. This follows from Theorem 3.7.9 and Lemma 3.7.10. □

3.8 Transformation into Canonical Form

We have remarked earlier that it is possible to transform every recursive type into a recursive type in canonical form that is denotationally equal under the standard denotation of “taking the tree unfolding”. We show in this section that there exists an effective method for transforming an arbitrary recursive type $\tau \in \mu Tp$

Figure 3.4: Inductive definition of a transformation $(\cdot)^c$ from μTp to $can\text{-}\mu Tp$.

τ	τ^c	
$\bot$	$\bot$	
$\top$	$\top$	
α	α	
$\tau_1 \to \tau_2$	$\tau_1^c \to \tau_2^c$	
$\mu\alpha.\,\tau_0$	τ_0^c	$\ldots$ if $\alpha \notin \mathrm{fv}(\tau_0^c)$
	$\bot$	$\ldots$ if $\tau_0^c \equiv \alpha$
	$\mu\alpha.\tau_0^c$	$\ldots$ if $\tau_0^c \equiv \rho_1 \to \rho_2$ and $\alpha \in \mathrm{fv}(\tau_0^c)$ (for some $\rho_1, \rho_2 \in \mu Tp$)
	$\mu\gamma.\,(\rho_1 \to \rho_2)[\gamma/\alpha][\gamma/\beta]$	$\ldots$ if $\tau_0^c \equiv \mu\beta.\,(\rho_1 \to \rho_2)$ (for some $\beta \in TVar$ and $\rho_1, \rho_2 \in \mu Tp$) and $\alpha \in \mathrm{fv}(\tau_0^c)$, and γ is the first variable in $TVar$ such that γ is substitutible in $\rho_1 \to \rho_2$ for both α and β, and $\big(\gamma \notin \mathrm{fv}(\rho_1 \to \rho_2)$ or $\gamma \equiv \alpha$ or $\gamma \equiv \beta\big)$

into a recursive type $\tau^c \in can\text{-}\mu Tp$ with the same tree unfolding, i.e. such that $\mathbf{Tree}(\tau) = \mathbf{Tree}(\tau^c)$ holds, and hence such that τ and σ are strongly equivalent. For this we define an effective transformation $(\cdot)^c : \mu Tp \to \mu Tp$, and subsequently demonstrate that it has the following two properties: firstly, $(\cdot)^c$ maps every recursive type τ to a recursive type τ^c in canonical form, and secondly, $(\cdot)^c$ maps every recursive type τ to a recursive type τ^c that is strongly equivalent with τ.

Although some important steps towards the proof of this main assertion about the transformation $(\cdot)^c$ will be given here, we are not going to present a direct proof (which would also be possible) at this stage. Instead, we defer one step of the proof, which relies on the soundness theorem of a later defined axiom system for $=_\mu$, to Chapter 5 where this proof system will be defined.

Now we define the transformation $(\cdot)^c$ right away.

Definition 3.8.1 (The function $(\cdot)^c$). The function

$$(\cdot)^c : \mu Tp \;\to\; \mu Tp\,, \quad \tau \;\mapsto\; \tau^c$$

is defined, for all recursive types $\tau \in \mu Tp$, by induction on the syntactical depth $|\tau|$ of τ, according to the clauses gathered in Figure 3.4.

In item (ii) of the lemma below the first desired property of $(\cdot)^c$ is shown: it is indeed a function that maps recursive types to recursive types in canonical form.

Lemma 3.8.2. *(i) Let $\tau, \sigma \in$ can-μTp and $\alpha \in TVar$ be such that σ is substitutible for α in τ. Then also $\tau[\sigma/\alpha] \in$ can-μTp follows.*

(ii) For all $\tau \in \mu Tp$ it holds that $\tau^c \in$ can-μTp. Consequently the range of $(\cdot)^c$ is contained in can-μTp.

Proof. (a) Part (i) is shown by a straightforward induction on the syntactical depth $|\tau|$ of τ.

(b) Also (ii) is shown by induction on $|\tau|$. Here all cases are immediate except the single one in which $\tau \equiv \mu\alpha.\,\tau_0$, $\tau_0^c \equiv \mu\beta.\,(\rho_1 \to \rho_2)$, and $\alpha \in \text{fv}(\tau_0^c)$ for some $\alpha, \beta \in TVar$ and $\tau, \tau_0, \rho_1, \rho_2 \in \mu Tp$. Since $|\tau_0| < |\tau|$, it follows by the induction hypothesis that $\tau_0^c \in$ *can-μTp*. A look at the grammar in Definition 3.1.1, (ii), which generates *can-μTp*, makes it clear that then also $\rho_1, \rho_2 \in$ *can-μTp*, and that $\beta \in \text{fv}(\rho_1 \to \rho_2)$ must hold. Now applications of item (i) of the lemma give that for $i = 1, 2$ first $\rho_i[\gamma/\alpha] \in$ *can-μTp* and then $\rho_i[\gamma/\alpha, \gamma/\beta] \equiv \rho_i[\gamma/\alpha][\gamma/\beta] \in$ *can-μTp* follows, where γ is the first variable in $TVar$ such that $(\gamma \notin \text{fv}(\rho_1 \to \rho_2) \vee \gamma \equiv \alpha \vee \gamma \equiv \beta)$ and γ is substitutible for both α and β in $\rho_1 \to \rho_2$. Since both α, β occur free in $\rho_1 \to \rho_2$, γ does so in $\rho_1[\gamma/\alpha, \gamma/\beta] \to \rho_2[\gamma/\alpha, \gamma/\beta] \equiv (\rho_1 \to \rho_2)[\gamma/\alpha, \gamma/\beta]$. Hence, with the definition of τ^c in the case considered here, $\tau^c \equiv \mu\gamma.\,(\rho_1 \to \rho_2)[\gamma/\alpha, \gamma/\beta] \equiv$
$\equiv \mu\gamma.\,(\rho_1[\gamma/\alpha, \gamma/\beta] \to \rho_2[\gamma/\alpha, \gamma/\beta]) \in$ *can-μTp*.

□

Extending Lemma 3.8.2, (ii), the following theorem states that the function $(\cdot)^c$ has indeed also the second of the two desired properties: it maps a recursive type τ to a recursive type τ^c that has the same tree unfolding as τ.

Theorem 3.8.3 (Desired properties of $(\cdot)^c$). *The function $(\cdot)^c : \mu Tp \to$ can-μTp defined in Definition 3.8.1 maps every recursive type τ to a* strongly equivalent *recursive type τ^c in canonical form, i.e. it holds:*

$$(\forall\, \tau \in \mu Tp)\ \left[\tau^c \in \textit{can-}\mu Tp \ \ \& \ \ \tau =_\mu \tau^c\right]. \tag{3.38}$$

A *proof* of this theorem will only be given on page 104 in Chapter 5. However, we formulate and prove a slightly weaker statement here, Lemma 3.8.4 below, the application of which will constitute an important step in the proof of Theorem 3.8.3 given later.

For the formulation of this lemma, we denote by $(\mu\mu{-}\mu)$ the following scheme of axioms

$$(\mu\mu{-}\mu)\quad \overline{\mu\alpha.\,\mu\beta.\,\tau = \mu\gamma.\,\tau[\gamma/\alpha, \gamma/\beta]} \quad \big(\text{if } (\alpha, \beta \in \text{fv}(\tau)\ \&\ \alpha \not\equiv \beta)\ \&$$
$$\&\ (\gamma \notin \text{fv}(\tau) \ \vee\ \gamma \equiv \alpha \ \vee\ \gamma \equiv \beta)\big)$$

for a proof system with μTp–Eq as its set of formulas.

Lemma 3.8.4. *For all $\tau \in \mu Tp$, the formula $\tau = \tau^c$ is provable in the extension* $\mathbf{WEQ} + (\mu\mu\text{–}\mu)$ *of the system* **WEQ** *by adding the axioms of the scheme* $(\mu\mu\text{–}\mu)$. *More formally, the statement*

$$\vdash_{\mathbf{WEQ}+(\mu\mu\text{–}\mu)} \tau = \tau^c$$

holds for all $\tau \in \mu Tp$.

Proof. By induction on the $|\tau|$. It is clear from a look at the table in Definition 3.8.1 that for $|\tau| = 0$ the formula $\tau = \tau^c$ is an axiom (REFL) of **WEQ**.

Only two cases shall be shown for the induction step if $|\tau| > 0$. The other cases can be settled similarly, or even much easier.

Suppose that $\tau \equiv \mu\alpha.\,\tau_0$ and $\tau_0^c \equiv \alpha$. Then by definition $\tau^c \equiv \bot$. By the induction hypothesis there exists a derivation $\mathcal{D}_0$ in $\mathbf{WEQ} + (\mu\mu\text{–}\mu)$ without assumptions and with conclusion $\tau_0 = \tau_0^c$; we choose such a derivation $\mathcal{D}_0$. Then the derivation $\mathcal{D}$

$$\dfrac{\mu\text{-COMPAT}\;\dfrac{\begin{matrix}\mathcal{D}_0\\ \tau_0 = \underbrace{\tau_0^c}_{\equiv\,\alpha}\end{matrix}}{\underbrace{\mu\alpha.\,\tau_0}_{\equiv\,\tau} = \mu\alpha.\,\alpha} \qquad \dfrac{(\mu-\bot)}{\mu\alpha.\,\alpha = \underbrace{\bot}_{\equiv\,\tau^c}}}{\tau = \tau^c}\;\text{TRANS}$$

is a derivation in $\mathbf{WEQ} + (\mu\mu\text{–}\mu)$ without assumptions and with conclusion $\tau = \tau^c$.

Suppose that $\tau = \mu\alpha.\,\tau_0$ and $\tau_0^c \equiv \mu\beta.\,(\rho_1 \to \rho_2)$ with $\alpha \in \mathrm{fv}(\tau_0^c)$. Then τ^c is defined as $\tau^c \equiv \mu\gamma.\,(\rho_1 \to \rho_2)[\gamma/\alpha, \gamma/\beta]$ for γ in *TVar* minimal such that ($\gamma \notin \mathrm{fv}(\rho_1 \to \rho_2)$ or $\gamma \equiv \alpha$ or $\gamma \equiv \beta$) and γ is substitutible in $\rho_1 \to \rho_2$ for α and β. By the induction hypothesis there exists a derivation $\mathcal{D}_0$ in $\mathbf{WEQ} + (\mu\mu-\mu)$ without assumptions and with conclusion $\tau_0 = \tau_0^c$; we choose such a derivation $\mathcal{D}_0$. From this the derivation $\mathcal{D}$ in $\mathbf{WEQ} + (\mu\mu-\mu)$ of the form

$$\text{TRANS}\;\dfrac{\mu\text{-COMPAT}\;\dfrac{\begin{matrix}\mathcal{D}_0\\ \tau_0 = \underbrace{\tau_0^u}_{\equiv\,\mu\beta.\,(\rho_1\to\rho_2)}\end{matrix}}{\underbrace{\mu\alpha.\,\tau_0}_{\equiv\,\tau} = \mu\alpha.\,\mu\beta.\,(\rho_1 \to \rho_2)} \qquad \dfrac{(\mu\mu-\mu)}{\mu\alpha.\,\mu\beta.\,(\rho_1 \to \rho_2) = \underbrace{\mu\gamma.\,(\rho_1 \to \rho_2)[\gamma/\alpha, \gamma/\beta]}_{\equiv\,\tau^c}}}{\tau = \tau^c}$$

can be built: $\mathcal{D}$ has the conclusion $\tau = \tau^c$ and does not contain assumptions.

□

Concluding this section, we want to remark that the statement

$$(\forall\, \tau \in \mu Tp)\;\; [\, \tau =_{w\mu} \tau^c \,]\,, \tag{3.39}$$

which results by replacing strong equivalence $=_{\mu}$ in (3.38) by weak equivalence $=_{w\mu}$ (and by dropping the part $\tau^c \in \mathit{can}\text{-}\mu Tp$ of (3.38) that is guaranteed by the statement in by Lemma 3.8.2, (ii)), *is* actually *not true*. Due to Lemma 3.8.4, (3.39) holds if and only if the formulas of the scheme $(\mu\mu{-}\mu)$ are theorems of **WEQ**. But this latter assertion is false, as can be shown by a very similar argument as used in the proof of Lemma 3.7.10.

3.9 Generated Subterms

An important fact about recursive types consists in the following assertion: for every recursive type σ there exist, up to taking variants, only finitely many recursive types that can be reached from σ by successive applications of the operations "renaming of bound variables", "unfolding at the outermost position", and "decomposition of a composite type $\chi_1 \to \chi_2$ into χ_1 or χ_2". This fact plays a vital part for showing completeness of coinductively motivated proof systems for recursive types ([BrHe98], for these systems see Chapter 5), and it will also be used for some of the proof-theoretic transformations developed in later chapters.

In this section we introduce, with the aim of a formalization of the mentioned statement, the notion of "generated subterm" of a recursive type. "Generated subterms" in our formalization are called "syntactical subterms" in [BrHe98]; in Remark 3.9.3 below we explain why we have chosen a different name for this notion here. We will gather some basic properties of this concept, among which the most important is the assertion, referred to above, that every recursive type has, up to variants, only a finite number of "generated subterms". And at the end of this section we will consider a certain extension of the notion "generated subterm" that will be useful later. Not all of the proofs for assertions in this section are given here. Some rather more tedious proofs for a couple of technical lemmas are deferred to Appendix A, Section A.5.

We are going to define, for all $\sigma \in \mu Tp$, a "generated subterm" of σ to be a 'more-step reduct' of σ with respect to a reduction relation $\to_{\text{roud}}$ on μTp that is the union of the three reduction relations $\to_{\text{ren}}$ for renaming bound variables, $\to_{\text{out-unf}}$ for unfolding a recursive type at its outermost position, and $\to_{\text{out-dec}}$ for decomposing a recursive type at its outermost position. For the purpose of this stipulation for "generated subterms", we first introduce the reduction relations $\to_{\text{out-unf}}$, $\to_{\text{out-dec}}$, and $\to_{\text{roud}}$ in a precise manner.

Definition 3.9.1 (The reduction relations $\to_{\text{out-unf}}$, $\to_{\text{out-dec}}$, $\to_{\text{roud}}$). The reduction relations $\to_{\text{out-unf}}$, $\to_{\text{out-dec}}$, and $\to_{\text{roud}}$ on μTp are defined as the following sets of pairs contained in $\mu Tp \times \mu Tp$:

$$\begin{aligned}
\to_{\text{out-unf}} &=_{\text{def}} \{\, \langle \mu\alpha.\, \tau,\ \tau[\mu\alpha.\, \tau/\alpha] \rangle \mid \alpha \in TVar,\ \tau \in \mu Tp \,\}\ , \\
\to_{\text{out-dec}} &=_{\text{def}} \{\, \langle \tau_1 \to \tau_2,\ \tau_i \rangle \mid \tau_1, \tau_2 \in \mu Tp,\ i \in \{0,1\} \}\ , \\
\to_{\text{roud}} &=_{\text{def}}\ \to_{\text{ren}} \cup \to_{\text{out-unf}} \cup \to_{\text{out-dec}}\ .
\end{aligned}$$

In the definition of $\rightarrow_{\mathrm{roud}}$ we have used the single-step renaming reduction $\rightarrow_{\mathrm{ren}}$ from Definition 3.4.1. ☒

We can now proceed to define "generated subterms" of a recursive type in the way as outlined prior to the definition above. We furthermore introduce notation for $\equiv_{\mathrm{ren}}$-equivalence classes on μTp, for the set of "generated subterms" of a recursive type, and for the set of $\equiv_{\mathrm{ren}}$-equivalence classes of generated subterms of a recursive type.

Definition 3.9.2 (Generated subterms, $\equiv_{\mathrm{ren}}$-equivalence classes of generated subterms). In the following three items, we define the notion "generated subterm of a recursive type" (this is done in (i)), and introduce some related terminology and notation.

(i) Let σ be a recursive type. For all $\tau \in \mu Tp$, we say that τ is a $\rightarrow_{\mathrm{roud}}$*-generated subterm* of σ, for short a *generated subterm* of σ, if and only if τ is a reduct of σ with respect to the reduction relation $\rightarrow_{\mathrm{roud}}$, i.e. iff $\sigma \twoheadrightarrow_{\mathrm{roud}} \tau$ holds. For all $\tau \in \mu Tp$, we denote by $\tau \sqsubseteq \sigma$ the assertion that τ is a generated subterm of σ , in accordance with the next stipulation: the *generated-subterm relation* $\sqsubseteq \subseteq \mu Tp \times \mu Tp$ is the inverse relation of $\twoheadrightarrow_{\mathrm{roud}}$, and more formally, it is defined, for all $\tau, \sigma \in \mu Tp$, by

$$\tau \sqsubseteq \sigma \quad \Longleftrightarrow_{\mathrm{def}} \quad \sigma \twoheadrightarrow_{\mathrm{roud}} \tau \ .$$

Furthermore, we denote by $\mathcal{G}(\sigma)$ the set of generated subterms of σ, i.e. we let

$$\mathcal{G}(\sigma) =_{\mathrm{def}} \{\tau \in \mu Tp \mid \tau \sqsubseteq \sigma\} \ ;$$

we also call $\mathcal{G}(\sigma)$ the *r̲enaming-o̲utermost-u̲nfolding-outermost-d̲ecomposition closure of* σ, for short the *roud-closure of* σ.

(ii) For all $\rho \in \mu Tp$, the notation $[\rho]_{\equiv_{\mathrm{ren}}}$ will be used for the equivalence class of ρ with respect to the variant relation $\equiv_{\mathrm{ren}}$ on μTp, i.e. we set more formally

$$[\rho]_{\equiv_{\mathrm{ren}}} =_{\mathrm{def}} \{\rho' \mid \rho' \in \mu Tp,\ \rho' \equiv_{\mathrm{ren}} \rho\} \ .$$

The quotient set $\mu Tp/_{\equiv_{\mathrm{ren}}} =_{\mathrm{def}} \big\{ [\rho]_{\equiv_{\mathrm{ren}}} \mid \rho \in \mu Tp \big\}$ of μTp with respect to $\equiv_{\mathrm{ren}}$ is the set of $\equiv_{\mathrm{ren}}$-equivalence-classes on μTp.

(iii) Let $\sigma \in \mu Tp$. The set $\mathcal{G}_*(\sigma)$ of equivalence classes with respect to $\equiv_{\mathrm{ren}}$ of generated subterms of σ, also called the *roud$_*$-closure of* σ, is defined by

$$\mathcal{G}_*(\sigma) =_{\mathrm{def}} \big\{ [\tau]_{\equiv_{\mathrm{ren}}} \mid \tau \sqsubseteq \sigma \big\} \ .$$

☒

Remark 3.9.3. Recursive types τ that are the result of a finite number of successive applications of one of the three operations "renaming", "outermost-unfolding" or "outermost-decomposition" to a recursive type σ are called "syntactical subterms" of σ in [BrHe98] (or, in [Br97], even just "subterms"). We have chosen not to use the expression "syntactical subterm of σ" for such recursive types τ, but speak of the *generated subterms of* σ instead, for the following reason: there does not exist a set-theoretical inclusion in either direction between the set of subterms and the set of "generated subterms" of a recursive type. It is neither always the case that a subterm of a recursive type σ is a "generated subterm" of σ, nor that a "generated subterm" of some $\sigma \in \mu Tp$ is also a subterm of σ.

However, there is nevertheless a certain flavour of "subterm" about "generated subterms", as the following two facts indicate:

1. Every recursive type τ is both a subterm and a generated subterm of itself.
2. If τ is a subterm, or respectively a generated subterm, of a recursive type σ, then τ is also a subterm, or respectively a generated subterm, of the recursive types $\sigma \to \chi$ and $\chi \to \sigma$, for all $\chi \in \mu Tp$.

We will be able to restate this observation below after introducing a proof system for the generated-subterm relation $\sqsubseteq$; then we will be able to compare this proof system with a similar one for the subterm relation $\trianglelefteq$ between recursive types.

A first observation about generated subterms of a recursive type is that both of the operations $\mathcal{G}(\cdot)$ and $\mathcal{G}_*(\cdot)$ on recursive types do not distinguish between different variants of a recursive type.

Proposition 3.9.4. *If two recursive types σ and σ' are variants of each other, then they possess the same generated subterms, i.e. for all $\sigma, \sigma' \in \mu Tp$ it holds: If $\sigma \equiv_{ren} \sigma'$, then $\mathcal{G}(\sigma) = \mathcal{G}(\sigma')$ and $\mathcal{G}_*(\sigma) = \mathcal{G}_*(\sigma')$ hold.*

Proof. Due to the fact that $\to_{\text{ren}}$ is a symmetrical reduction relation, it follows that

$$\sigma \equiv_{\text{ren}} \sigma' \quad \Longleftrightarrow \quad \sigma \twoheadrightarrow_{\text{ren}} \sigma'$$

holds for all $\sigma, \sigma' \in \mu Tp$. As a consequence, it holds for all $\sigma, \sigma', \tau \in \mu Tp$ that

$$\sigma \equiv_{\text{ren}} \sigma' \implies \big(\sigma \twoheadrightarrow_{\text{roud}} \tau \iff \sigma' \twoheadrightarrow_{\text{roud}} \tau \big) .$$

Thus, if σ and σ' are variants of each other, then they have the same generated subterms. □

Furthermore, it is obvious from the definition of the generated-subterm relation $\sqsubseteq$ as the inverse relation $\twoheadleftarrow_{\text{roud}}$ of the more-step reduction relation $\twoheadrightarrow_{\text{roud}}$ with respect to $\to_{\text{roud}}$ that $\sqsubseteq$ is reflexive and transitive. It is easy to verify that $\sqsubseteq/_{\equiv_{\text{ren}}} \subseteq \mu Tp/_{\equiv_{\text{ren}}} \times \mu Tp/_{\equiv_{\text{ren}}}$, the quotient relation of $\sqsubseteq$ with respect to $\equiv_{\text{ren}}$, is well-defined by the clause

$$[\tau]_{\equiv_{\text{ren}}} \sqsubseteq/_{\equiv_{\text{ren}}} [\sigma]_{\equiv_{\text{ren}}} \quad \Longleftrightarrow_{\text{def}} \quad \tau \sqsubseteq \sigma \qquad (\text{for all } \tau, \sigma \in \mu Tp),$$

and that $\sqsubseteq/_{\equiv_{\text{ren}}}$ is again reflexive and transitive. However, neither of the relations $\sqsubseteq$ and $\sqsubseteq/_{\equiv_{\text{ren}}}$ is anti-symmetric, which can be seen, for example, from the two recursive types $\tau \equiv \mu\alpha.\,(\bot \to (\top \to \alpha))$ and $\sigma \equiv \top \to \tau$, for which $\tau \not\equiv_{\text{ren}} \sigma$ and $\tau \to_{\text{roud}} \bot \to (\top \to \tau) \to_{\text{roud}} \sigma$ hold; hence, in particular, τ is a generated subterm of σ and σ is a generated subterm of τ.

Example 3.9.5. (i) Considering the recursive type $\mu\alpha\beta.\,\gamma$, we notice that the only reduction sequence from this recursive type that avoids $\to_{\text{ren}}$-steps is

$$\mu\alpha\beta.\,\gamma \to_{\text{out-unf}} \mu\beta.\,\gamma \to_{\text{out-unf}} \gamma\,. \tag{3.40}$$

It is obvious that additional $\to_{\text{ren}}$-steps do not enable $\to_{\text{roud}}$-reduction sequences leading to recursive types that are not variants of one of the three types in (3.40). Hence we find

$$\mathcal{G}(\mu\alpha\beta.\,\gamma) = [\mu\alpha\beta.\,\gamma]_{\equiv_{\text{ren}}} \cup [\mu\beta.\,\gamma]_{\equiv_{\text{ren}}} \cup \{\gamma\}\,,$$
$$\mathcal{G}_*(\mu\alpha\beta.\,\gamma) = \big\{[\mu\alpha\beta.\,\gamma]_{\equiv_{\text{ren}}}\,, [\mu\beta.\,\gamma]_{\equiv_{\text{ren}}}\,, \{\gamma\}\big\}$$

for the roud-closure and the roud$_*$-closure of $\mu\alpha\beta.\,\gamma$.

(ii) Every $\to_{\text{ren}}$-avoiding reduction from $\mu\alpha\beta\gamma\delta.\,\gamma$ is an initial segment of the reduction

$$\begin{aligned}\mu\alpha\beta\gamma\delta.\,\gamma \to_{\text{out-unf}} \mu\beta\gamma\delta.\,\gamma \to_{\text{out-unf}} \mu\gamma\delta.\,\gamma \to_{\text{out-unf}}& \\ \to_{\text{out-unf}} \mu\delta\gamma\delta.\,\gamma \to_{\text{out-unf}} \mu\gamma\delta.\,\gamma \to_{\text{out-unf}} \cdots\,,&\end{aligned} \tag{3.41}$$

which enters a loop of length two after two reduction steps. Additional $\to_{\text{ren}}$-steps do not lead to generated subterms of $\mu\alpha\beta\gamma\delta.\,\gamma$ that are not variants of one of the four recursive types in (3.41). Hence we conclude

$$\begin{aligned}\mathcal{G}(\mu\alpha\beta\gamma\delta.\,\gamma) &= [\mu\alpha\beta\gamma\delta.\,\gamma]_{\equiv_{\text{ren}}} \cup [\mu\beta\gamma\delta.\,\gamma]_{\equiv_{\text{ren}}} \cup [\mu\gamma\delta.\,\gamma]_{\equiv_{\text{ren}}} \cup [\mu\delta\gamma\delta.\,\gamma]_{\equiv_{\text{ren}}}\,,\\ \mathcal{G}_*(\mu\alpha\beta\gamma\delta.\,\gamma) &= \big\{[\mu\alpha\beta\gamma\delta.\,\gamma]_{\equiv_{\text{ren}}}\,, [\mu\beta\gamma\delta.\,\gamma]_{\equiv_{\text{ren}}}\,, [\mu\gamma\delta.\,\gamma]_{\equiv_{\text{ren}}}\,, [\mu\delta\gamma\delta.\,\gamma]_{\equiv_{\text{ren}}}\big\}\,.\end{aligned}$$

(iii) For the recursive type $\sigma_1 \equiv \mu\alpha.\,((\alpha \to \bot) \to \top)$, we find the $\to_{\text{ren}}$-step-free reduction sequences

$$\underbrace{\mu\alpha.\,((\alpha \to \bot) \to \top)}_{\equiv\,\sigma_1} \;\to_{\text{out-unf}}\; ((\sigma_1 \to \bot) \to \top) \;\to_{\text{out-dec}}$$
$$\to_{\text{out-dec}} \begin{cases} (\sigma_1 \to \bot) & \to_{\text{out-dec}} \begin{cases} \sigma_1 & \to_{\text{out-unf}} \;\cdots \\ \bot \end{cases} \\ \top \end{cases}$$

where we recognize a loop of length three. Furthermore, it is quite obvious that extending any of these reduction sequences by $\to_{\text{ren}}$-steps and other $\to_{\text{roud}}$-steps does not lead to generated subterms of σ_1 that are not variants of any of the five subterms of σ_1 as encountered above. Hence we can conclude for the roud-closure and the roud$_*$-closure of σ_1:

$$\mathcal{G}(\sigma_1) = \big\{\sigma_1',\, (\sigma_1' \to \bot) \to \top,\, \sigma_1' \to \bot,\, \bot,\, \top \mid \sigma_1' \in \mu Tp,\, \sigma_1' \equiv_{\text{ren}} \sigma_1\big\}\,,$$
$$\mathcal{G}_*(\sigma_1) = \big\{\,[\sigma_1]_{\equiv_{\text{ren}}}\,, [\sigma_1 \to \bot]_{\equiv_{\text{ren}}}\,, [(\sigma_1 \to \bot) \to \top]_{\equiv_{\text{ren}}}\,, \{\bot\}, \{\top\}\,\big\}\,.$$

(iv) Concerning the recursive type $\sigma_2 \equiv \mu\alpha.\,(\beta \to \mu\beta.\,(\alpha \to \beta))$, we observe the following $\to_{\text{roud}}$-steps:

$$\left.\begin{array}{l} \underbrace{\mu\alpha.\,(\beta \to \mu\beta.\,(\alpha \to \beta))}_{\equiv\,\sigma_2} \;\to_{\text{ren}}\; \underbrace{\mu\alpha.\,(\beta \to \mu\gamma.\,(\alpha \to \gamma))}_{\equiv\,\sigma_2'} \;\to_{\text{out-unf}} \\ \qquad \to_{\text{out-unf}} (\beta \to \mu\gamma.\,(\sigma_2' \to \gamma)) \;\to_{\text{out-dec}}\; \left\{\begin{array}{l} \beta \\ \mu\gamma.\,(\sigma_2' \to \gamma)\,, \end{array}\right. \end{array}\right\} \tag{3.42}$$

where the $\to_{\text{ren}}$-step was necessary to facilitate the subsequent $\to_{\text{out-unf}}$-step, and

$$\underbrace{\mu\gamma.\,(\sigma_2' \to \gamma)}_{\equiv\,\chi} \;\to_{\text{out-unf}}\; \sigma_2' \to \chi \;\to_{\text{out-dec}}\; \left\{\begin{array}{lll} \sigma_2' & \to_{\text{out-unf}} & \cdots \\ \chi & \to_{\text{out-unf}} & \cdots\,. \end{array}\right\} \tag{3.43}$$

We find two loops in the reductions described by (3.42) and (3.43), of length five and another one of length two. It is easy to see that additional applications of $\to_{\text{ren}}$, followed by other $\to_{\text{roud}}$-reduction steps, do not increase the set of $\equiv_{\text{ren}}$-equivalence classes of generated subterms occurring in one of the reduction sequences as sketched above. Hence we find in particular:

$$\begin{aligned} \mathcal{G}_*(\sigma_2) = \{\, & [\sigma_2]_{\equiv_{\text{ren}}}\,,\, [\beta \to \mu\gamma.\,(\sigma_2' \to \gamma)]_{\equiv_{\text{ren}}}\,, \\ & \{\beta\},\, [\chi]_{\equiv_{\text{ren}}}\,,\, [\sigma_2' \to \chi]_{\equiv_{\text{ren}}} \,\}\,. \end{aligned} \tag{3.44}$$

$\blacksquare$

For the purpose of making it possible to prove assertions about the generated-subterm relation $\boldsymbol{\sqsubseteq}$ by proof-theoretic means, we introduce the following formal system.

Definition 3.9.6 (The axiom system gST). The formal system **gST**, a (pure) Hilbert system, possesses as its *formulas* precisely all expressions $\tau \sqsubseteq \sigma$ with arbitrary $\tau, \sigma \in \mu Tp$.[7] Its *axioms* are all those of the form (REFL) in Figure 3.5 and its *inference rules* are the rules REN$_r$, FOLD$_r$, and B→CTXT$_r$, applications of which are gathered and described as respective inference schemes in Figure 3.5. ⊠

The name of the rule B→CTXT$_r$ of **gST** is motivated by the operation on formulas of **gST** that it allows to perform: to replace the recursive type τ on the right-hand side of a formula by the substitution $C[\tau]$ of τ into a "basic →context" $C \in \mu Tp\text{–}Ctxt$ of the form $\Box \to \sigma$ or $\sigma \to \Box$, for some $\sigma \in \mu Tp$.

We can now give the following characterization of the generated-subterm relation $\boldsymbol{\sqsubseteq}$ in terms of derivability in **gST**.

[7] Note the difference between the symbol $\sqsubseteq$ used in formulas of **gST** and its boldface-version $\boldsymbol{\sqsubseteq}$ that denotes the generated-subterm relation according to Definition 3.9.2 (the difference between the symbols $\sqsubseteq$ and $\boldsymbol{\sqsubseteq}$ is better visible in the font size used for the text).

Figure 3.5: The axiom system **gST** for the generated-subterm relation $\sqsubseteq$ on μTp.

The *axioms* of **gST** :

$$\text{(REFL)}\ \overline{\tau \sqsubseteq \tau}$$

The *inference rules* of **gST** :

$$\frac{\tau \sqsubseteq \sigma'}{\tau \sqsubseteq \sigma}\ \text{REN}_r \quad (\text{if } \sigma \equiv_{\text{ren}} \sigma') \qquad\qquad \frac{\tau \sqsubseteq \sigma[\mu\beta.\,\sigma/\beta]}{\tau \sqsubseteq \mu\beta.\,\sigma}\ \text{FOLD}_r$$

$$\frac{\tau \sqsubseteq \sigma_j}{\tau \sqsubseteq \sigma_1 \to \sigma_2}\ \text{B}{\to}\text{CTXT}_r \quad (\text{where } j \in \{1,2\})$$

Proposition 3.9.7 (Characterization of generated subterms in terms of derivability in gST). *For all $\tau, \sigma \in \mu Tp$ it holds that*

$$\tau \textit{ is a generated subterm of } \sigma \quad \Longleftrightarrow \quad \vdash_{\mathbf{gST}} \tau \sqsubseteq \sigma\ . \tag{3.45}$$

Hint on the Proof. The assertion of the lemma can be reformulated as

$$(\forall \tau, \sigma \in \mu Tp)\ \big[\ \sigma \twoheadrightarrow_{\text{roud}} \tau \quad \Longleftrightarrow \quad \vdash_{\mathbf{gST}} \tau \sqsubseteq \sigma\ \big]\ . \tag{3.46}$$

The logical implication "⇒" within (3.46) can be shown by a straightforward induction on the length of $\to_{\text{roud}}$-reduction sequences, and the implication "⇐" by an easy induction on the depth of derivations in **gST** with conclusion $\tau = \sigma$ and without assumptions.

Remark 3.9.8. The axiomatization **gST** for the generated-subterm relation $\sqsubseteq$ on μTp is similar to the proof system **SUBT**, given in Figure 3.6, that can easily be recognized to be sound and complete with respect to the subterm relation $\trianglelefteq$ on μTp (as this has been defined in Definition 3.2.3), i.e. for which

$$(\forall\, \tau, \sigma \in \mu Tp)\ \big[\ \ \tau \trianglelefteq \sigma \quad \Longleftrightarrow \ \vdash_{\mathbf{SUBT}}\ \tau \trianglelefteq \sigma\ \ \big]$$

holds. An obvious similarity between the systems **gST** and **SUBT** consists in the presence in both systems of the axioms (REFL), and of the analogous respective basic-→-context rules $\text{B}{\to}\text{CTXT}_r$ (this fact is a reformulation of the observations (I) and (II) in Remark 3.9.3). However, these systems differ with respect to rules involving μ-terms as the recursive types on the right-hand side in the conclusions of their instances: whereas **gST** allows folding of a recursive type at the right-hand side of a formula, **SUBT** admits the formation of a "basic μ-context".

For the purpose of proving the main theorem about generated subterms of recursive types, we formulate a number of technical lemmas all of which are proved

Figure 3.6: The axiom system **SUBT** for the subterm relation $\trianglelefteq$ on μTp.

The *axioms* of **SUBT**:

$$\text{(REFL)}\ \frac{}{\tau \trianglelefteq \tau}$$

The *inference rules* of **SUBT**:

$$\frac{\tau \trianglelefteq \sigma_j}{\tau \trianglelefteq \sigma_1 \to \sigma_2}\ \text{B}{\to}\text{CTXT}_r \quad (\text{for } j \in \{1,2\}) \qquad\qquad \frac{\tau \trianglelefteq \sigma}{\tau \trianglelefteq \mu\beta.\,\sigma}\ \text{B}\mu\text{CTXT}_r$$

in Appendix A, Section A.5. There, all of these lemmas are demonstrated by proof-theoretical arguments, exploiting the characterization (3.45) of the generated-subterm relation $\sqsubseteq$ via derivability in the formal system **gST**.

The first lemma states, for all recursive types σ, that every variable that occurs freely in σ is a generated subterm of σ, and that the free variables of generated subterms of σ are contained among the free variables of σ.

Lemma 3.9.9. *(i) For all $\tau, \sigma \in \mu Tp$ it holds: $\tau \in \mathcal{G}(\sigma) \Rightarrow fv(\tau) \subseteq fv(\sigma)$.*

(ii) For all $\sigma \in \mu Tp$ and $\alpha \in TVar$ it holds: $\alpha \in fv(\sigma) \Rightarrow \alpha \in \mathcal{G}(\sigma)$.

The *proof* of this lemma is given in Appendix A, Section A.5, on page 341.

The second lemma gives a representation of the set of generated subterms of $\sigma[\rho/\alpha]$, for all $\tau, \sigma, \rho \in \mu Tp$ and $\alpha \in TVar$, in terms of the generated subterms of σ and ρ.

Lemma 3.9.10. *Let $\sigma, \rho \in \mu Tp$ and $\alpha \in TVar$ such that ρ is substitutible for α in ρ. Then for the set $\mathcal{G}(\sigma[\rho/\alpha])$ of subterms of $\sigma[\rho/\alpha]$ it holds that*

$$\mathcal{G}(\sigma[\rho/\alpha]) \subseteq \{\tau \mid \tau \in \mu Tp,\ \tau \equiv_{ren} \chi[\rho/\alpha],\ \chi \in \mathcal{G}(\sigma)\} \cup \mathcal{G}(\rho)\ .$$

The *proof* of this lemma uses Lemma 3.9.9 and is given in Appendix A, Section A.5, on page 345. It depends on a technical statement, Lemma A.5.1, in the same section of Appendix A, about the possible forms of derivations in **gST** without assumptions and with conclusion $\tau \sqsubseteq \sigma[\rho/\alpha]$, where $\tau, \sigma, \rho, \in \mu Tp$ and $\alpha \in TVar$.

The last lemma needed for the main theorem on generated subterms gives respective representations of the sets of generated subterms of a recursive type σ in terms of the sets of generated subterms of ‘immediate’ subterms of σ, by which we mean subterms of σ from which σ has been built in the last formation step according to the grammar (3.1).

Lemma 3.9.11. *Let $\sigma, \sigma_0, \sigma_1, \sigma_2 \in \mu Tp$ and $\beta \in TVar$. Then the following three statements hold for the set $\mathcal{G}(\sigma)$ of generated subterms of σ:*

(i) $\mathcal{G}(\sigma) = \{\sigma\}$ for all $\sigma \in \{\bot, \top\} \cup TVar$.

(ii) $\mathcal{G}(\sigma_1 \to \sigma_2) \;=\; [\sigma_1 \to \sigma_2]_{\equiv_{ren}} \cup \mathcal{G}(\sigma_1) \cup \mathcal{G}(\sigma_2)\,.$

(iii) $\mathcal{G}(\mu\beta.\,\sigma) \;\subseteq\; [\mu\beta.\,\sigma]_{\equiv_{ren}} \cup \big\{\rho \mid \rho \equiv_{ren} \chi[\mu\beta.\,\sigma/\beta],\, \chi \in \mathcal{G}(\sigma)\big\}\,.$

The *proof* for this lemma, which uses the assertion of Lemma 3.9.10 for the demonstration of item (iii), can be found in Appendix A, Section A.5, on page 345.

Eventually, we are able to state and prove, by using just Lemma 3.9.11, the main theorem about generated subterms.

Theorem 3.9.12. *Every recursive type has only a finite number of generated subterms if variants of each other are not counted separately. That is, the set $\mathcal{G}_*(\sigma)$ is finite for all $\sigma \in \mu Tp$. And what is more, the bound*

$$|\mathcal{G}_*(\sigma)| \le s(\sigma) \qquad (\textit{for all } \sigma \in \mu Tp\,) \tag{3.47}$$

holds, i.e. for all $\sigma \in \mu Tp$ the cardinality of $\mathcal{G}_*(\sigma)$ is bounded by the size of σ.

Proof. Obviously, it suffices to show (3.47). We will prove this statement by using Lemma 3.9.11 in a proof by induction on the depth $|\sigma|$ of a recursive type σ.

For the base case of the induction, let $\sigma \in \mu Tp$ be such that $|\sigma| = 0$. Then it must be the case that $\sigma \in \{\bot, \top\} \cup TVar$. By Lemma 3.9.11, (i), we know that $\mathcal{G}(\sigma) = \{\sigma\}$. Since there is no occurrence of a μ-binding in σ, also $\mathcal{G}_*(\sigma) = \{\{\sigma\}\}$ follows. Hence clearly $|\mathcal{G}_*(\sigma)| = 1 = \mathrm{s}(\sigma)$.

For the treatment of the induction step, let $\sigma \in \mu Tp$ such that $|\sigma| > 0$. We distinguish two cases, according to whether σ starts with a μ-operator or not.

If σ does not start with a μ-binder, σ is of the form $\sigma_1 \to \sigma_2$ for some $\sigma_1, \sigma_2 \in \mu Tp$. By Lemma 3.9.11, (ii), it follows

$$\mathcal{G}_*(\sigma) = \mathcal{G}_*(\sigma_1 \to \sigma_2) = \{[\sigma_1 \to \sigma_2]_{\equiv_{\mathrm{ren}}}\} \cup \mathcal{G}_*(\sigma_1) \cup \mathcal{G}_*(\sigma_2)\;.$$

Hence, by this and the use of the induction hypothesis, we obtain

$$\begin{aligned} |\mathcal{G}_*(\sigma)| \;&\le\; 1 + |\mathcal{G}_*(\sigma_1)| + |\mathcal{G}_*(\sigma_2)| \\ &\le\; 1 + \mathrm{s}(\sigma_1) + \mathrm{s}(\sigma_2) = \mathrm{s}(\sigma)\;. \end{aligned}$$

In the second case, σ is of the form $\mu\beta.\,\sigma_0$ for some $\sigma_0 \in \mu Tp$ and $\beta \in TVar$. Here

$$\mathcal{G}_*(\mu\beta.\,\sigma_0) \;\subseteq\; \{\,[\mu\beta.\,\sigma_0]_{\equiv_{\mathrm{ren}}}\,\} \cup \big\{\,[\,\chi\,[\mu\beta.\,\sigma_0/\beta]\,]_{\equiv_{\mathrm{ren}}} \mid \chi \in \mathcal{G}(\sigma_0)\big\} \tag{3.48}$$

follows as a consequence of Lemma 3.9.11, (iii). Since the assertion associated with (3.18) in Lemma 3.4.2 implies that, for all $\chi_1, \chi_2 \in \mu Tp$ such that $\mu\beta.\,\sigma_0$ is substitutible for β in χ_1 and χ_2,

$$\chi_1 \equiv_{\mathrm{ren}} \chi_2 \;\;\Longrightarrow\;\; [\,\chi_1\,[\mu\beta.\,\sigma_0/\beta]\,]_{\equiv_{\mathrm{ren}}} = [\,\chi_2\,[\mu\beta.\,\sigma_0/\beta]\,]_{\equiv_{\mathrm{ren}}}$$

holds, it follows

$$|\big\{\,[\chi\,[\mu\beta.\,\sigma_0/\beta]]_{\equiv_{\mathrm{ren}}} \mid \chi \in \mathcal{G}(\sigma_0)\big\}| \;\le\; |\mathcal{G}_*(\sigma_0)|\;. \tag{3.49}$$

Using (3.48), (3.49), and the induction hypothesis, we eventually find:

$$\begin{aligned} |\mathcal{G}_*(\sigma)| = |\mathcal{G}_*(\mu\beta.\,\sigma_0)| &\le 1 + |\mathcal{G}_*(\sigma_0)| \\ &\le 1 + \mathrm{s}(\sigma_0) = \mathrm{s}(\mu\beta.\,\sigma_0) = \mathrm{s}(\sigma)\ . \end{aligned}$$

This concludes the induction step and thus the proof of the theorem. □

As a corollary to Theorem 3.9.12, we are now able to prove Fact 3.5.11 from Section 3.5, that is, the assertion that the tree unfolding of a recursive type is a regular (type) tree.

Corollary 3.9.13. *For every recursive type σ, its tree unfolding* **Tree***(σ) is a regular tree, i.e.* **Tree***(σ) has only finitely many subtrees. Moreover, for every $\sigma \in \mu Tp$, the number of different subtrees of the tree unfolding* **Tree***(σ) of σ is limited by the size $s(\sigma)$ of σ.*

Sketch of Proof. We perform the proof in the following two items, where the proof of a statement in the first item is merely sketched.

(a) It is straightforward to check from the clauses (1)–(5) in Definition 3.5.8 that the inductive definition of $\mathbf{Tree}(\sigma)(p)$, for arbitrary $\sigma \in \mu Tp$ and paths $p \in \{0,1\}^*$, recurs on the definition of $\mathbf{Tree}(\sigma_0)(p_0)$ *only* for such $\sigma_0 \in \mu Tp$ and $p_0 \in \{0,1\}^*$, for which $|p_0| \le |p|$ and σ_0 is a generated subterm of σ. This implies that, for all $\sigma \in \mu Tp$ and $p \in \{0,1\}^*$ for which $\mathbf{Tree}(\sigma)(p)$ is in fact defined, a generated subterm τ of σ with the property $\mathbf{Tree}(\sigma)|_p = \mathbf{Tree}(\tau)$ can always be found effectively by expanding the definition of $\mathbf{Tree}(\sigma)(p)$. More precisely, the statement

$$\begin{aligned} &(\forall \sigma \in \mu Tp)\,(\forall p \in \{0,1\}^*)\ \big[\,\mathbf{Tree}(\sigma)(p)\downarrow\ \Longrightarrow \\ &\qquad\qquad \Longrightarrow\ (\exists\, \tau \in \mathcal{G}(\sigma))\,[\,\mathbf{Tree}(\sigma)|_p = \mathbf{Tree}(\tau)\,]\,\big] \qquad (3.50) \end{aligned}$$

can be shown by demonstrating the statement in (3.50) without the two leading quantifications, for all pairs $(\sigma,p) \in \mu Tp \times \{0,1\}^*$, by induction on pairs $(|p|, \mathrm{nl}\mu\mathrm{b}(\sigma))$ in $\omega \times \omega$ with respect to the lexicographic ordering on this set, where $|p|$ means the length of the string p and $\mathrm{nl}\mu\mathrm{b}(\sigma)$ is the number of the leading μ-bindings in σ as defined in Definition 3.5.4 (phrased differently the proof amounts, as this was the case with Definition 3.5.8, to the use of induction on the length of p together with a subinduction on $\mathrm{nl}\mu\mathrm{b}(\sigma)$).

(b) Since, for all $\sigma \in \mu Tp$, an arbitrary subtree of $\mathbf{Tree}(\sigma)$ is of the form $\mathbf{Tree}(\sigma)|_p$ for some path $p \in \{0,1\}^*$, we can now conclude the following from statement (3.50): For arbitrary $\sigma \in \mu Tp$ there can be at most as many different subtrees of the tree unfolding $\mathbf{Tree}(\sigma)$ as there are different generated subterms of σ. Since by Lemma 3.5.10 the operation **Tree** of producing the tree unfolding of a recursive type has the same result $\mathbf{Tree}(\tilde{\sigma})$ for all variants $\tilde{\sigma}'$ of a recursive type $\tilde{\sigma}$, it follows that the number of different subtrees of $\sigma \in \mu Tp$

is even limited by $|\mathcal{G}_*(\sigma)|$, the number of the $\equiv_{\text{ren}}$-equivalence-classes of generated subterms of σ. But from the theorem we know that $|\mathcal{G}_*(\sigma)| \leq \mathrm{s}(\sigma)$ for all $\sigma \in \mu Tp$, and hence the statement of the corollary follows.

□

In Lemma 3.9.11 we have encountered a representation of the set of generated subterms of a recursive type σ in terms of the sets of generated subterms of immediate subterms of σ. Extending and sharpening this statement, we now also give the following theorem that provides, for all recursive types σ that fulfill the variable condition *VC*, an explicit description of the set of generated subterms of σ via subterm occurrences in σ.

Theorem 3.9.14. (Representation of the generated subterms of a recursive type via its subterms). *Let σ be a recursive type that fulfills the variable condition VC. Then every generated subterm of σ is a variant of the recursive type denoted by the (according to Convention 3.3.6 admissible) substitution expression*

$$\rho\,[\mu\beta_n.\,\sigma_n/\beta_n]\dots[\mu\beta_1.\,\sigma_1/\beta_1]\;, \tag{3.51}$$

where it holds that

$$\left.\begin{aligned} & n\in\omega \;\;\&\;\; \rho,\sigma_1,\dots,\sigma_n\in\mu Tp \;\;\&\;\; \beta_1,\dots,\beta_n\in TVar \;\;\& \\ & (\exists\, p,p_1,\dots,p_n\in Pos(\sigma))\,\big[\,\rho=\sigma|_p \;\;\&\;\; p_1<\dots<p_n<p \;\;\& \\ & \qquad\qquad \&\;\; (Pref(p)\backslash\{p\})\cap\mu Pos(\sigma)=\{p_1,\dots,p_n\} \;\;\& \\ & \qquad\qquad \&\;\; (\forall i\in\{1,\dots,n\})\,[\,\sigma|_{p_i}=\mu\beta_i.\,\sigma_i\,]\,\big]\;, \end{aligned}\right\} \tag{3.52}$$

i.e. where ρ is a subterm of σ ($\rho \trianglelefteq \sigma$ holds) that occurs at some position p in σ, and where the recursive types $\mu\beta_1.\,\sigma_1,\dots,\mu\beta_n.\,\sigma_n$ are the occurrences in σ of μ-terms at positions $p_1,\dots,p_n$ above position p, listed from the topmost such occurrence down to the bottommost one (see Figure 3.7 for an illustration). It follows that the roud$_*$-closure of σ can be written as

$$\mathcal{G}_*(\sigma)=\left\{\Big[\rho\,[\mu\beta_n.\,\sigma_n/\beta_n]\dots[\mu\beta_1.\,\sigma_1/\beta_1]\Big]_{\equiv_{ren}} \;\middle|\; (3.52)\ \textit{holds}\right\}. \tag{3.53}$$

Before turning to the proof of this theorem, we consider an example for its application.

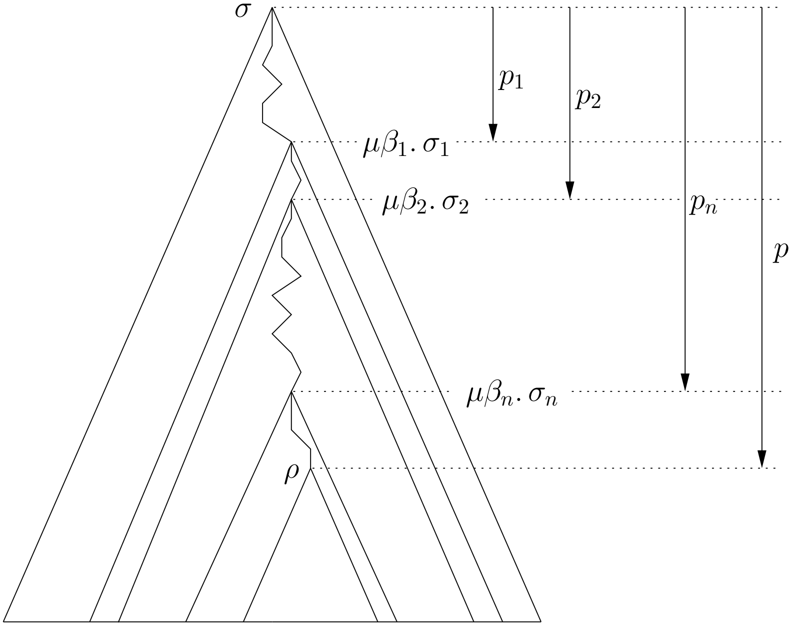
Example 3.9.15. We consider the recursive type

$$\sigma \;\equiv\; \mu\alpha.\,(\beta\to\mu\gamma.\,(\alpha\to\gamma))\;,$$

which is the recursive type σ'_2 that we encountered in Example 3.9.5, (iv). Quite obviously, σ fulfills the variable condition *VC*. From the picture of the term tree of σ in Figure 3.8, we can read the set of positions of σ and the set of positions of μ-expressions of σ, namely,

$$Pos(\sigma)=\{\epsilon,1,11,12,121,1211,1212\}\;, \quad\text{and}\qquad \mu Pos(\sigma)=\{\epsilon,12\}\;.$$

Figure 3.7: Illustration of condition (3.52) in Theorem 3.9.14: For each subterm occurrence $\rho = \sigma|_p$ in σ, the subterms $\mu\beta_1.\,\sigma_1, \ldots, \mu\beta_n.\,\sigma_n$ denote all occurrences in σ, at positions of increasing length $p_1 < \ldots < p_n$ above p, of subterms starting with a μ-binding.



This entails that the set of subterms of σ is of the form:

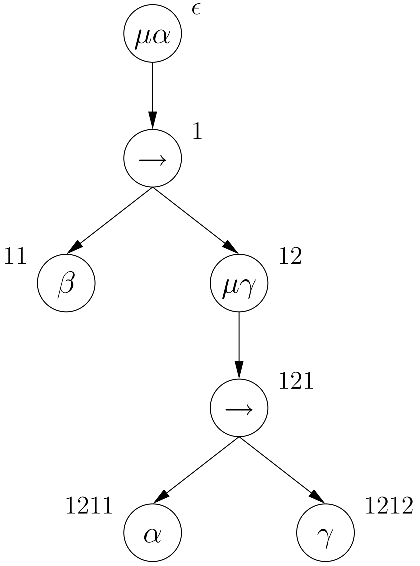
$$\mathrm{Subt}(\sigma) = \{\sigma,\, \beta \to \mu\gamma.\,(\alpha \to \gamma),\, \beta,\, \mu\gamma.\,(\alpha \to \gamma),\, \alpha \to \gamma,\, \alpha,\, \gamma\} \;.$$

With this, Theorem 3.9.14 now implies that every generated subterm of σ is a variant of a recursive type that is one of the forms

$$\begin{aligned}
&\sigma|_\epsilon \;\equiv\; \sigma \;, \\
&\sigma|_1[\sigma|_\epsilon/\alpha] \;\equiv\; \beta \to \mu\gamma.\,(\alpha \to \gamma)[\sigma/\alpha] \;\equiv\; \beta \to \mu\gamma.\,(\sigma \to \gamma) \;, \\
&\sigma|_{11}[\sigma|_\epsilon/\alpha] \;\equiv\; \beta[\sigma/\alpha] \;\equiv\; \beta \;, \\
&\sigma|_{12}[\sigma|_\epsilon/\alpha] \;\equiv\; \mu\gamma.\,(\alpha \to \gamma)[\sigma/\alpha] \;\equiv\; \mu\gamma.\,(\sigma \to \gamma) \;, \\
&\sigma|_{121}[\sigma|_{12}/\gamma][\sigma|_\epsilon/\alpha] \;\equiv\; (\alpha \to \gamma)[\mu\gamma.\,(\alpha \to \gamma)/\gamma][\sigma/\alpha] \;\equiv\; \sigma \to \mu\gamma.\,(\sigma \to \gamma) \;, \\
&\sigma|_{1211}[\sigma|_{12}/\gamma][\sigma|_\epsilon/\alpha] \;\equiv\; \alpha[\mu\gamma.\,(\alpha \to \gamma)/\gamma][\sigma/\alpha] \;\equiv\; \sigma \;, \\
&\sigma|_{1212}[\sigma|_{12}/\gamma][\sigma|_\epsilon/\alpha] \;\equiv\; \gamma[\mu\gamma.\,(\alpha \to \gamma)/\gamma][\sigma/\alpha] \;\equiv\; \mu\gamma.\,(\alpha \to \gamma) \;.
\end{aligned}$$

In this list we have encountered two recursive types twice, which entails that the

Figure 3.8: The positions in the recursive type $\sigma \equiv \mu\alpha.\,(\beta \to \mu\gamma.\,(\alpha \to \gamma))$.



roud$_*$-closure of σ consists of the following five-element set:

$$\mathcal{G}_*(\sigma) = \{[\sigma]_{\equiv_{\text{ren}}}\,,\ [\beta \to \mu\gamma.\,(\sigma \to \gamma)]_{\equiv_{\text{ren}}}\,,\ [\beta]_{\equiv_{\text{ren}}}\,,\\ [\mu\gamma.\,(\sigma \to \gamma)]_{\equiv_{\text{ren}}}\,,\ [\sigma \to \mu\gamma.\,(\sigma \to \gamma)]_{\equiv_{\text{ren}}}\}\ .$$

This implies that the roud-closure of σ is

$$\mathcal{G}(\sigma) = \big\{\,\sigma',\ \beta \to \mu\gamma'.\,(\sigma' \to \gamma'),\ \beta,\\ \mu\gamma'.\,(\sigma' \to \gamma'),\ \sigma' \to \mu\gamma'.\,(\sigma' \to \gamma')\ \big|\ \gamma' \in \mathit{TVar},\ \sigma' \equiv_{\text{ren}} \sigma\,\big\}\ .$$

We note that what we have found here by an application of Theorem 3.9.14 conforms with the representation (3.44) of $\mathcal{G}(\sigma_2')$ for the variant σ_2 of σ: in view of Proposition 3.9.4, $\mathcal{G}(\sigma)$ here and $\mathcal{G}(\sigma_2)$ in (3.44) should agree, as they in fact do because of $\sigma_2 \equiv_{\text{ren}} \sigma$ and $\sigma \equiv \sigma_2'$. ■

Our proof of Theorem 3.9.14 consists of two lemmas that are stated below. These lemmas use a reduction relation $\to_{\text{oud}}$, the $\to_{\text{ren}}$-free part of the reduction relation $\to_{\text{roud}}$, as an auxiliary concept, as well as the notions "$\to_{\text{oud}}$-generated subterms", "oud-closure", and "oud$_*$-closure" of a recursive type that are induced by $\to_{\text{oud}}$.

Hereby, the reduction relation $\to_{\text{oud}} \subseteq \mu Tp \times \mu Tp$ is defined as

$$\to_{\text{oud}}\ \ =_{\text{def}}\ \ \to_{\text{out-unf}} \cup \to_{\text{out-dec}}\ ,$$

i.e. as the union of the reduction relations $\to_{\text{out-unf}}$ and $\to_{\text{out-dec}}$. Analogously as the reduction relation $\to_{\text{roud}}$ induced the notions of $\to_{\text{roud}}$-generated subterm, the roud-closure, and the roud$_*$-closure of a recursive type, the reduction relation $\rightarrow_{\text{oud}}$ induces the notions "$\rightarrow_{\text{oud}}$-generated subterm", "oud-closure", and "oud$_*$-closure" of a recursive type.

For their definition, let σ be a recursive type. We say that $\tau \in \mu Tp$ is a *$\rightarrow_{oud}$-generated subterm of* σ if and only if $\sigma \twoheadrightarrow_{\text{oud}} \tau$ holds. And furthermore, we define the *oud-closure* $\tilde{\mathcal{G}}(\sigma)$ *of* σ and the *oud$_*$-closure* $\tilde{\mathcal{G}}_*(\sigma)$ *of* σ by

$$\tilde{\mathcal{G}}(\sigma) =_{\text{def}} \{\tau \mid \sigma \twoheadrightarrow_{\text{oud}} \tau\}\ , \quad \text{and by} \qquad \tilde{\mathcal{G}}_*(\sigma) =_{\text{def}} \left\{ [\tau]_{\equiv_{\text{ren}}} \mid \sigma \twoheadrightarrow_{\text{oud}} \tau \right\}\ .$$

Relying on these definitions, we now give two lemmas that will together constitute a proof of Theorem 3.9.14.

Lemma 3.9.16. *Let σ be a recursive type that fulfills the variable condition VC. Then every $\rightarrow_{oud}$-generated subterm of σ is of the form*

$$\rho\,[\mu\beta_n.\,\sigma_n/\beta_n]\ldots[\mu\beta_1.\,\sigma_1/\beta_1] \tag{3.54}$$

where $n \in \omega$, $\rho, \sigma_1, \ldots, \sigma_n$ and $\beta_1, \ldots, \beta_n \in TVar$ such that assertion (3.52) *in Theorem 3.9.14 holds. It follows that the oud-closure $\tilde{\mathcal{G}}(\sigma)$ and the oud$_*$-closure $\tilde{\mathcal{G}}_*(\sigma)$ of σ can be written as*

$$\tilde{\mathcal{G}}(\sigma) = \left\{\rho\,[\mu\beta_n.\,\sigma_n/\beta_n]\ldots[\mu\beta_1.\,\sigma_1/\beta_1] \mid (3.52)\ \textit{holds}\right\}\ ,\ \textit{and} \tag{3.55}$$

$$\tilde{\mathcal{G}}_*(\sigma) = \left\{ \left[\rho\,[\mu\beta_n.\,\sigma_n/\beta_n]\ldots[\mu\beta_1.\,\sigma_1/\beta_1]\right]_{\equiv_{ren}} \mid (3.52)\ \textit{holds}\right\}\ . \tag{3.56}$$

A *proof* of this lemma can be found in Appendix A, Section A.5, on page 350.

Lemma 3.9.17. *Let σ be a recursive type that fulfills the variable condition VC_0, i.e. let $\sigma \in \mu Tp$ be such that $VC_0(\sigma)$ is the case. Then $\mathcal{G}_*(\sigma) = \tilde{\mathcal{G}}_*(\sigma)$ holds.*

A *proof* for this lemma can again be found in Appendix A, Section A.5, on page 355. – We can now give the proof for Theorem 3.9.14.

Proof of Theorem 3.9.14. Let σ be an arbitrary recursive type that fulfills the variable condition *VC*. Then σ clearly also fulfills the variable condition VC_0. Hence both Lemma 3.9.16 and Lemma 3.9.17 are applicable. The first lemma implies the representation (3.56) for the oud$_*$-closure $\tilde{\mathcal{G}}_*(\sigma)$ of σ, and the second lemma entails that the oud$_*$-closure $\tilde{\mathcal{G}}_*(\sigma)$ of σ is equal to the roud$_*$-closure $\mathcal{G}_*(\sigma)$ of σ. Hence

$$\mathcal{G}_*(\sigma) = \left\{ \left[\rho\,[\mu\beta_n.\,\sigma_n/\beta_n]\ldots[\mu\beta_1.\,\sigma_1/\beta_1]\right]_{\equiv_{\text{ren}}} \mid (3.52)\ \text{holds}\right\} \tag{3.57}$$

follows. In this way we have shown assertion (3.53) of the theorem. This entail also the first assertion of the theorem, namely, that every generated subterm of σ is of the form (3.51), which is also an admissible substitution expression, due to Lemma 3.9.16. □

As an immediate corollary to Theorem 3.9.14, we can now prove the special case of the assertion of Theorem 3.9.12 that states that the roud$_*$-closure is finite for all recursive types that fulfill the variable condition *VC*.

Corollary 3.9.18. *Let σ be a recursive type that fulfills the variable condition VC. Then*

$$|\mathcal{G}_*(\sigma)| \leq s(\sigma)$$

holds, i.e. the cardinality of $\mathcal{G}_*(\sigma)$ is bounded by the size of σ.

Proof. Let σ be an arbitrary recursive type that fulfills the variable condition *VC*. Then by Theorem 3.9.14 the representation (3.53) holds for the roud$_*$-closure $\mathcal{G}_*(\sigma)$ of σ. Thus an element of $\mathcal{G}_*(\sigma)$ is always a $\equiv_{\text{ren}}$-equivalence class of the form

$$\big[\rho\,[\mu\beta_n.\,\sigma_n/\beta_n]\ldots[\mu\beta_1.\,\sigma_1/\beta_1]\big]_{\equiv_{\text{ren}}}\ , \tag{3.58}$$

where ρ is a subterm of σ, and where for ρ, n, and $\mu\beta_1.\,\sigma_1,\ldots,\mu\beta_n.\,\sigma_n$ the assertion (3.52) holds (and hence where $\mu\beta_1.\,\sigma_1,\ldots,\mu\beta_n.\,\sigma_n$ depend on what particular subterm-occurrence of ρ in σ is considered). From this

$$|\mathcal{G}_*(\sigma)| \leq |\mathit{Pos}(\sigma)| = \mathrm{s}(\sigma)$$

follows, by using Proposition 3.2.4. □

And by using this corollary, we are also able to give the following alternative proof for Theorem 3.9.12.

Alternative Proof for Theorem 3.9.12. Let σ be an arbitrary recursive type. We choose a variant σ' of σ that fulfills the variable condition *VC*. By Proposition 3.9.4 we find $\mathcal{G}_*(\sigma) = \mathcal{G}_*(\sigma')$. Using this, by Corollary 3.9.18 it follows that

$$|\mathcal{G}_*(\sigma)| = |\mathcal{G}_*(\sigma')| \leq \mathrm{s}(\sigma') = \mathrm{s}(\sigma)\ ,$$

where we have also used the easy observation that the size of a recursive type is invariant under going over to a variant. □

With the aim of easing the formulation of some assertions later on, it is useful to extend the generated-subterm relation $\sqsubseteq$ to a more general generated-subterm relation $\sqsubseteq'$ that allows to view the bottom-type $\perp$ as a more general form of generated subterm of recursive types like $\mu\alpha.\,\alpha$ and $\mu\alpha_1\ldots\alpha_n.\,\alpha_1$ (that have the same tree unfolding as $\perp$), and like $\beta \to \mu\gamma\delta.\,\gamma$ (which has $\mu\gamma\delta.\,\gamma$ as generated subterm that has the same tree unfolding as $\perp$). For defining $\sqsubseteq'$, we need to define an extension $\to_{\text{roud}\perp}$ of the reduction relation $\to_{\text{roud}}$ from Definition 3.9.1 by a reduction relation $\to_{\text{out-}(\mu\text{-}\perp)'}$ that reduces recursive types $\mu\alpha_1\ldots\alpha_n.\,\alpha_1$ (where $\alpha_1,\ldots,\alpha_n \in \mathit{TVar}$), which have the same tree unfolding as $\perp$, to $\perp$.

Definition 3.9.19 (The reduction relations $\to_{\text{out-}(\mu\text{-}\perp)'}$, $\to_{\text{roud}\perp}$). The reduction relations $\to_{\text{out-}(\mu\text{-}\perp)'}$ and $\to_{\text{roud}\perp}$ on $\mu\mathit{Tp}$ are defined as the following sets contained in $\mu\mathit{Tp}\times\mu\mathit{Tp}$:

$$\begin{aligned}\to_{\text{out-}(\mu\text{-}\perp)'} &=_{\text{def}} \{\,\langle\mu\alpha_1\ldots\alpha_n.\,\alpha_1,\ \perp\rangle \mid n\in\omega\backslash\{0\}, \alpha_1\ldots\alpha_n\in\mathit{TVar}\,\}\ ,\\ \to_{\text{roud}\perp} &=_{\text{def}}\ \to_{\text{roud}}\,\cup \to_{\text{out-}(\mu\text{-}\perp)'}\end{aligned}$$

(for defining $\to_{\text{roud}\perp}$ we have used here the reduction relation $\to_{\text{roud}}$ from Definition 3.9.1). ⊠

Relying on the reduction relation $\to_{\mathrm{roud}\bot}$, we proceed to define "$\to_{\mathrm{roud}\bot}$-generated subterms" of a recursive type σ as reducts of σ with respect to $\to_{\mathrm{roud}\bot}$. We also introduce notation for the set of generated subterms, and for the set of $\equiv_{\mathrm{ren}}$-equivalence classes of a generated subterm, of a recursive type.

Definition 3.9.20 ($\to_{\mathbf{roud}\bot}$-generated subterms, $\equiv_{\mathbf{ren}}$-equivalence classes of $\to_{\mathbf{roud}\bot}$-generated subterms). Let σ be a recursive type.

(i) For all $\tau \in \mu Tp$, we say that τ is a $\to_{\mathrm{roud}\bot}$-*generated subterm* of σ if and only if τ is a more-step reduct of σ with respect to the reduction relation $\to_{\mathrm{roud}\bot}$, i.e. iff $\sigma \twoheadrightarrow_{\mathrm{roud}\bot} \tau$ holds. We call the relation $\sqsubseteq' \subseteq \mu Tp \times \mu Tp$, defined as the inverse relation $\twoheadleftarrow_{\mathrm{roud}\bot}$ of $\twoheadrightarrow_{\mathrm{roud}\bot}$, the $\to_{\mathrm{roud}\bot}$-*generated-subterm relation* on μTp. And we denote by $\mathcal{G}'(\sigma)$ the set of generated subterms of σ, i.e. we let

$$\mathcal{G}'(\sigma) =_{\mathrm{def}} \{\tau \in \mu Tp \mid \tau \sqsubseteq' \sigma\} ;$$

we also call $\mathcal{G}'(\sigma)$ the *roud⊥-closure of* σ.

(ii) Let $\sigma \in \mu Tp$. The set $\mathcal{G}'_*(\sigma)$ of equivalence classes with respect to $\equiv_{\mathrm{ren}}$ of $\to_{\mathrm{roud}\bot}$-generated subterms of σ, also called the *roud$\bot_*$-closure of* σ, is defined by

$$\mathcal{G}'_*(\sigma) =_{\mathrm{def}} \left\{ [\tau]_{\equiv_{\mathrm{ren}}} \mid \tau \sqsubseteq' \sigma \right\} .$$

⊠

With the aim of enabling a similar logical formalization of the relation $\sqsubseteq'$ as the axiomatization **gST** that we introduced for the generated-subterm relation $\sqsubseteq$, we extend **gST** to the following formal system **gST**$'$.

Definition 3.9.21 (The proof system gST$'$). The proof system **gST**$'$ is the extension of the formal system **gST**, given in Definition 3.9.6, by adding the *inference rule* $(\mu\!-\!\bot)_r^{\bot\mathrm{der}}$ with applications of the form

$$\frac{\tau \sqsubseteq \bot}{\tau \sqsubseteq \mu\alpha_1 \ldots \alpha_n.\,\alpha_1}\ (\mu\!-\!\bot)_r^{\bot\mathrm{der}}$$

(for all $\tau \in \mu Tp$, $n \in \omega\backslash\{0\}$, and $\alpha_1, \ldots, \alpha_n \in \mu Tp$). ⊠

The following characterization of the $\to_{\mathrm{roud}\bot}$-generated-subterm relation $\sqsubseteq'$ can be proved similarly as explained in the hint given for the proof of Proposition 3.9.7.

Proposition 3.9.22 (Characterization of $\to_{\mathbf{roud}\bot}$-generated subterms via derivability in gST$'$). *For all $\tau, \sigma \in \mu Tp$ it holds*

$$\tau \textit{ is a } \to_{\mathit{roud}\bot}\textit{-generated subterm of } \sigma \quad \Longleftrightarrow \quad \vdash_{\mathbf{gST}'} \tau \sqsubseteq \sigma .$$

The next lemma asserts that, for all recursive types σ, a $\to_{\mathrm{roud}\bot}$-generated subterm of σ is either $\bot$ or a generated subterm of σ.

Lemma 3.9.23. *For all $\sigma \in \mu Tp$, the two following statements hold:*

$$\begin{aligned} \mathcal{G}'(\sigma) &\subseteq \mathcal{G}(\sigma) \cup \{\bot\}\,, \\ \mathcal{G}'_*(\sigma) &\subseteq \mathcal{G}_*(\sigma) \cup \{\{\bot\}\}\,. \end{aligned}$$

Proof. We observe that in a derivation in $\mathbf{gST}'$ only trivial applications of REN with premise and conclusion $\bot \sqsubseteq \bot$ can occur above an application of $(\mu{-}\bot)_r^{\bot\mathrm{der}}$. Hence every $\mathbf{gST}'$-derivation $\mathcal{D}$ with conclusion $\tau \sqsubseteq \sigma$ contains either zero or one application(s) of $(\mu - \bot)_r^{\bot\mathrm{der}}$; in the first case, $\mathcal{D}$ is also a $\mathbf{gST}$-derivation and hence τ is a generated subterm of σ, and in the second case, $\tau \equiv \bot$. From this the assertions of the lemma follow as a consequence of the characterization of the $\to_{\mathrm{roud}\bot}$-generated-subterm relation $\sqsubseteq'$ in Proposition 3.9.22. □

Using this lemma, we get the following corollary as an immediate consequence of Theorem 3.9.12.

Corollary 3.9.24. *Every recursive type has only a finite number of $\to_{\mathrm{roud}\bot}$-generated subterms if variants of each other are not counted separately. That is, the set $\mathcal{G}'_*(\sigma)$ is finite for all $\sigma \in \mu Tp$. And what is more, the bound*

$$|\mathcal{G}'_*(\sigma)| \leq s(\sigma) + 1 \qquad \textit{(for all } \sigma \in \mu Tp\textit{)} \tag{3.59}$$

holds, i.e. for all $\sigma \in \mu Tp$ the cardinality of $\mathcal{G}'_*(\sigma)$ is bounded by the size of σ plus one.

As an aside, we mention that the bound (3.59) is not optimal: it is not difficult to prove that also

$$|\mathcal{G}'_*(\sigma)| \leq \mathrm{s}(\sigma) \qquad \text{(for all } \sigma \in \mu Tp\text{)}$$

holds. This bound, however, is precise in the sense that equality holds in some cases: for example, let $\sigma \equiv \mu\alpha.\,\alpha$; then $\mathcal{G}'_*(\sigma) = \big\{\{\bot\}, [\mu\alpha.\,\alpha]_{\equiv_{\mathrm{ren}}}\big\}$ and hence $|\mathcal{G}'_*(\sigma)| = 2 = \mathrm{s}(\sigma)$ holds.

We conclude this section about generated subterms with a lemma that asserts a bound on the number of leading μ-bindings of $\to_{\mathrm{roud}\bot}$-generated subterms of a recursive type. We will need this lemma later in Appendix C (for the proof there of Theorem C.11).

Lemma 3.9.25 (A bound on $\mathbf{nl}\mu\mathbf{b}(\cdot)$ for $\to_{\mathbf{roud}\bot}$-generated subterms). *The number of leading μ-bindings in a $\to_{roud\bot}$-generated subterm of a recursive type σ is bounded by the double of the syntactical depth of σ. More formally, for all $\tau, \sigma \in \mu Tp$ the following implication holds:*

$$\tau \sqsubseteq' \sigma \quad\Longrightarrow\quad nl\mu b(\tau) \;\leq\; 2\,|\sigma|\,. \tag{3.60}$$

A *proof* for this lemma is given in Appendix A, Section A.5, on page 356.

Chapter 4

Derivability and Admissibility of Inference Rules

In this chapter we are concerned with the notions of derivability and admissibility of inference rules in formal systems. We give definitions and gather basic results for these notions with respect to abstract formulations of those sorts of formal systems under which the proof systems for recursive types that will be encountered in the next chapter can be subsumed: pure Hilbert systems and natural-deduction systems. The properties of rules to be derivable or admissible with respect to a formal system is introduced and studied for pure Hilbert systems and for natural-deduction systems separately in Section 4.2 and in Section 4.3. We start, however, by giving an informal explanation of the relevance of rule derivability and rule admissibility for the aim of finding proof-theoretic transformations in general; this is the topic of Section 4.1.

4.1 Relevance for the Construction of Proof-Transformations

As explained in the Chapter 1, our main aim consists in finding proof-theoretic transformations between proof systems for recursive type equality (which will be introduced in Chapter 5). In particular, we are interested in finding transformations between proof systems with the same set of formulas and the same set of theorems; however, our argumentation below applies also to the problem of finding transformations from a system $\mathcal{S}_1$ into a system $\mathcal{S}_2$ that is an extension of $\mathcal{S}_1$, i.e. where $\mathcal{S}_2$ has possibly a richer formula language and more theorems than $\mathcal{S}_1$. Still more precisely, we are interested in transformations of such kind that, in the case of considered proof systems $\mathcal{S}_1$ and $\mathcal{S}_2$, produce, for an arbitrary given derivation

$\mathcal{D}_1$ in $\mathcal{S}_1$, a derivation $\mathcal{D}_2$ in $\mathcal{S}_2$ such that $\mathcal{D}_2$ has the same conclusion as $\mathcal{D}_1$; and such that $\mathcal{D}_2$ demonstrates that its conclusion is a theorem of $\mathcal{S}_2$ whenever $\mathcal{D}_1$ witnesses that its conclusion is a theorem of $\mathcal{S}_1$ (this clearly presupposes that $\mathcal{S}_2$ is an extension of $\mathcal{S}_1$).

Undoubtedly, there are many different ways conceivable of how transformations of this kind may proceed and can be constructed. Nevertheless, it seems reasonable to distinguish two kinds of approaches for finding transformations that are helpful in many cases: the "rule translation method" and the "rule elimination method". In the two items below we give rough descriptions of each of these methods.

The Rule Translation Method. The approach prescribed by this method consists of three steps (RT1), (RT2), and (RT3), which are outlined subsequently. In step (RT1) show that, for every axiom A of $\mathcal{S}_1$, a derivation $\mathcal{D}^{(A)}$ in $\mathcal{S}_2$ can be found such that $\mathcal{D}^{(A)}$ demonstrates A to be a theorem of $\mathcal{S}_2$. Then, in step (RT2), show that all instances of rules in $\mathcal{S}_1$ can be 'translated' into 'simulating derivations' in $\mathcal{S}_2$ (we will later use the term "mimicking derivation" instead according to a definition in this chapter). Eventually, in step (RT3), a transformation of the desired kind can, under some assumptions on the 'well-behavedness' of the set of derivations in $\mathcal{S}_2$, be constructed as follows: for a given derivation $\mathcal{D}_1$ in $\mathcal{S}_1$, exchange all occurrences of axioms A in $\mathcal{D}_1$ by occurrences of the derivations $\mathcal{D}^{(A)}$ in $\mathcal{S}_2$, and replace all rule applications in $\mathcal{D}_1$ by respective 'simulating derivations' in $\mathcal{S}_2$; under the mentioned assumptions on $\mathcal{S}_2$, the result will be a derivation $\mathcal{D}_2$ in $\mathcal{S}_2$ with the same conclusion as $\mathcal{D}_1$. And finally, show that a derivation $\mathcal{D}_2$ found in this way witnesses that its conclusion is a theorem of $\mathcal{S}_2$ whenever the conclusion of $\mathcal{D}_1$ is demonstrated by $\mathcal{D}_1$ to be a theorem of $\mathcal{S}_1$.

The Rule Elimination Method. This method suggests the following approach to finding a proof-theoretic transformation from the derivations in a proof system $\mathcal{S}_1$ into derivations in a proof system $\mathcal{S}_2$. Try to proceed in the steps (RE1), (RE2), and (RE3) as detailed below. In step (RE1) show, as in the translation method, that the axioms of $\mathcal{S}_1$ can be derived as theorems in $\mathcal{S}_2$ by describing how to construct respective derivations $\mathcal{D}^{(A)}$ in $\mathcal{S}_2$ for all axioms A of $\mathcal{S}_1$. In step (RE2), demonstrate for each rule R in $\mathcal{S}_1$ that all of its applications can effectively be removed from an arbitrary derivation $\mathcal{D}$ in the extension $\mathcal{S}_2{+}R$ of $\mathcal{S}_2$ by adding the rule R with the result of a 'simulating derivation' $\mathcal{D}'$ in $\mathcal{S}_2$ that, in particular, has the same conclusion as $\mathcal{D}$ (for some purposes it will be sufficient to show this only for such derivations in $\mathcal{S}_2{+}R$ that witness that their conclusions are theorems). If this happens to be possible, then proceed, in step (RE3), to build a transformation from derivations in $\mathcal{S}_1$ into derivations in $\mathcal{S}_2$ that proceeds like this: in a given derivation $\mathcal{D}_1$ in $\mathcal{S}_1$ that demonstrates its conclusion to be a theorem of $\mathcal{S}_1$, first replace all occurrences of axioms A by simulating derivations $\mathcal{D}^{(A)}$ in $\mathcal{S}_2$; and then eliminate stepwisely, in a top-down manner, all applications of rules from $\mathcal{S}_1$ from the resulting derivation according to the eliminations found in step (RE2) (if such eliminations turn out to be not applicable, or lead to unwanted results, possibly step (RE2) has to be refined); show that in this way always a desired derivation $\mathcal{D}_2$ in $\mathcal{S}_2$ is found that has the same conclusion as $\mathcal{D}_1$ and that demonstrates this conclusion to be a theorem of $\mathcal{S}_2$.

In general, the rule translation method can be subsumed under the rule elimination method as a special case: finding translations of applications of a rule R from a system $\mathcal{S}_1$ into simulating derivations in a system $\mathcal{S}_2$ will most likely also define a way how such rule applications can be eliminated from derivations in the extension $\mathcal{S}_2{+}R$ of $\mathcal{S}_2$ by adding R (and in particular from such derivations that demonstrate their conclusions to be theorems). However, transformations that are based on the rule translation method are generally of a substantially easier kind: here a rule application can always be translated directly without taking the particular context into consideration in which it appears within a derivation (for transformations based on rule elimination procedures this is typically not the case). But there are cases in which the rule translation method cannot be applied, and where the rule elimination method still leads to the construction of a transformation. By this we mean the possibility of one of the following situations. Either the provable fact that not for all rules of a system $\mathcal{S}_1$ there exist translations into appropriate derivations of $\mathcal{S}_2$. Or the perhaps only temporary situation, encountered while trying to build a transformation between proof systems $\mathcal{S}_1$ and $\mathcal{S}_2$, that one does not succeed in giving translations of a particular rule R of $\mathcal{S}_1$ into derivations of $\mathcal{S}_2$, though such translations might yet exist, while one is already able to show by a more detailed proof-theoretic argument that applications of R can always be eliminated from theorem-demonstrating derivations in the extension $\mathcal{S}_2{+}R$ of $\mathcal{S}_2$ by adding R.

In many 'real' cases, however, a combination between the translation method and the elimination methods is called for. It is a quite frequent experience during attempts to find a transformation that some rules of a proof system $\mathcal{S}_1$ enable straightforward translations into $\mathcal{S}_2$, whereas for others only elimination procedures can be given for derivations in extensions of $\mathcal{S}_2$ (either because for their applications translations into $\mathcal{S}_2$ do actually not exist, or because such translations have simply not yet been found). Under these circumstances a proof-theoretic transformation has to be constructed in a more complicated way. For example, by initially 'simulating', in a given derivation in $\mathcal{S}_1$, applications of rules of the first kind by their translations into $\mathcal{S}_2$, and by subsequently eliminating those rules for which (at the moment or for inherent reasons) only elimination procedures exist.

Let us turn now to the question of how the two methods described above are related to the main topic of this chapter. Basically, an answer can be put as follows: there are respective close relationships between, on the one hand, the rule translation method and the notion of rule derivability, and on the other hand, between the rule elimination method and the notion of rule admissibility.

In the first case the connection can be explained as follows. A new rule R is generally called "derivable" (or "derived") with respect to a formal system $\mathcal{S}$ if its 'operational behaviour', i.e. the possibility R offers to produce certain conclusions when certain premises are given, can always be, in some sense, 'modeled', 'simulated', or 'mimicked' by appropriate derivations in $\mathcal{S}$; hence if applications of R can be 'translated' into derivations in $\mathcal{S}$. Therefore step (RT2) in the rule translation method can be restated, for given proof systems $\mathcal{S}_1$ and $\mathcal{S}_2$, as the step (RT2)$'$: show that all rules of $\mathcal{S}_1$ are *derivable* in $\mathcal{S}_2$ (with respect to a 'sensible' formulation of rule derivability in $\mathcal{S}_2$).

And in the second case, the connection is established in the following manner. A rule R is usually called "admissible" in a proof system $\mathcal{S}$ if and only if the admission of R as an additional rule to the rules of the system $\mathcal{S}$ does not lead to more theorems in the extended system $\mathcal{S}{+}R$. Therefore generally the following can be concluded: if a rule R is admissible in a system $\mathcal{S}$ then, for every derivation $\mathcal{D}$ in $\mathcal{S}{+}R$ that demonstrates its conclusion to be a theorem of $\mathcal{S}{+}R$, there exists a derivation $\mathcal{D}'$ in $\mathcal{S}$ that has the same conclusion as $\mathcal{D}$ and that witnesses this conclusion to be a theorem of $\mathcal{S}$. (By the word "generally" we refer to the 'natural' assumption on $\mathcal{S}$ and $\mathcal{S}{+}R$ that all theorems of these systems are demonstrated by respective derivations, and that derivations can only be used to demonstrate their conclusions as theorems.) Consequently, step (RE2) in the rule elimination method can be restated, for given proof systems $\mathcal{S}_1$ and $\mathcal{S}_2$, as the step (RE2)$'$: prove, for every rule R of $\mathcal{S}_1$, that R is *admissible* in $\mathcal{S}_2$.

Motivated by these connections with methods for finding proof-transformations, we will investigate, in (the following sections of) this chapter and in Appendix B, the notions of rule derivability and admissibility in abstract formulations of such sorts of proof systems that we will encounter later among proof systems for recursive type equality: pure Hilbert systems and natural-deduction systems. Our aim is to gather general results about this notions, and in particular, we are interested in what precise consequences the property of a rule R to be admissible or derivable in a system $\mathcal{S}$ has for the possibility to eliminate applications of R from derivations in the extension of $\mathcal{S}$ by adding R.

Rather than defining and studying rule derivability and admissibility only for the particular proof systems that will be introduced in Chapter 5, we have thereby decided to consider these notions in more abstract settings first. The reason why we prefer to proceed in this way is connected to the following fact. Definitions for when a rule R is to be called "admissible" or "derivable" with respect to a system $\mathcal{S}$ are usually given only with respect to specific classes of proof systems with a given concrete formula language and for a particular way how rules can be defined (rules are mostly defined schematically, using substitution on a meta-language of the formula language of a system). But a consequence of this is that some very general properties of these notions are, we feel, not normally made as clear as this is possible rather easily. This holds in particular for the relationship between rule derivability and admissibility and specific forms of how rule elimination is possible. These considerations have lead us to set out first for an investigation of rule derivability and admissibility in abstract formalizations of proof systems.

Another part of our treatment of rule derivability and admissibility in this chapter, namely Section 4.3, is motivated by the fact that these notions are usually not studied for natural-deduction-style systems. A possible reason for this is that formal definitions of rules and derivations are inherently more complex for natural-deduction-style systems than for Hilbert-style systems. However, in the course of our investigation concerning proof systems for recursive type equality, natural-deduction-style systems arise naturally in the shape of "Brandt-Henglein systems", which contain coinductively motivated rules for 'circular' reasoning. Therefore the aim of finding proof-transformations involving proof systems of this kind has lead us, via attempts to apply the rule translation and rule elimination methods, to questions concerning derivability and admissibility of inference rule in natural-deduction-style proof systems. The results in Section 4.3 and in Section B.2 of Appendix B have been motivated in this way.

In later chapters, the definitions and formal statements developed in this chapter (as well as in Appendix B) will serve mainly as background knowledge for the actual construction of proof-transformations. Nevertheless, they will be of some definite, albeit indirect, use. To sketch an example: we will come across situations where we succeed in giving a rather involved elimination procedure for applications of a rule R, belonging to a system $\mathcal{S}_1$, from derivations in the extension $\mathcal{S}_2+R$ of a system $\mathcal{S}_2$ by adding R; in other words, a situation in which we have shown admissibility of a rule R in a system $\mathcal{S}_2$ in some complicated, but nevertheless effective, way. In such circumstances we will frequently convince ourselves, by showing that R is not derivable in $\mathcal{S}_2$, that we have not overlooked some easy possibility to translate applications of R into derivations in $\mathcal{S}_2$. (But we will obviously not gain any more certainty by such an argument that a particular elimination procedure, we will have found, for R-applications in derivations of $\mathcal{S}_2+R$ cannot be simplified substantially.)

4.2 Definitions and Results in Pure Hilbert Systems

In this section we define rule derivability and admissibility in a subclass of Hilbert-style proof calculi, namely in the class of "pure Hilbert systems". The definitions here are of a certain informal character due to the fact that we do not base ourselves on a completely formalized concept of "proof system" belonging to this class (and of "inference rule" for such a system). We rather use rough descriptions of these systems. However, we will refer to the definitions and results in Section B.1 of Appendix B, where we introduce the concept "Abstract Pure Hilbert System" (APHS) and give precise definitions of rule derivability and admissibility in such systems. Also, we do not give proofs for the results gathered in this section, but we will again confer to Section B.1 of Appendix B, where for each proposition and theorem given here a corresponding "exact version" with respect to APHS's is formulated and where *some* proofs are given.

4.2.1 Formal Systems

The simplest and traditionally the most widely used proof systems in the literature on formal logic occur under a variety of names, with the terms "formal systems", "axiom(atic) systems" and "Hilbert(-style) systems for theoremhood" among them. For instance, in [Shoe67] Shoenfield uses the term "formal system" in the sense of formal axiom systems which he loosely describes as follows: “Every formal system contains as its parts a language, axioms, and rules of inference; its theorems are defined inductively from axioms and rules”. In a slightly more explicit way, in [Avr91] Avron describes a “formal system” in traditional understanding as containing the following components:

1. A formal language L with several syntactic categories, one of which is the category of ‘well-formed formulae’ (wff).
2. An effective set of wff called ‘axioms’.
3. An effective set of rules (called ‘inference rules’) for deriving theorems from the axioms.

And then, “the set of ‘theorems’ is usually taken to be the minimal set of wff which includes all the axioms and is closed under the rules of inference” ([Avr91]).

Rule applications in such systems are typically inferences of the form

$$\frac{A_1 \;\; \ldots \;\; A_n}{B} \tag{4.1}$$

where $n \in \omega$, $A_1, \ldots, A_n$ are formulas called the *premises* and B is a formula called the *conclusion* of this application. Hereby the case $n = 0$ of a rule application with no premises is included in contrast to applications with an infinite number of premises. Although rules usually allow only applications with a fixed number of premises, we will not use this restriction here. We will call rules with applications of the form (4.1) *pure Hilbert-system rules* (because rules of this form will be used below in the description of “pure Hilbert systems”, see Subsection 4.2.3 below).

Let $\mathcal{FS}$ be a formal system. By a ‘proof’ of a formula A in $\mathcal{FS}$ usually a finite sequence $\sigma = (A_1, \ldots, A_n)$ of formulas of $\mathcal{FS}$ is meant, where $A_n \equiv A$ and each formula in σ is either the occurrence of an axiom or that of a formula which is the conclusion of an application of a rule of $\mathcal{FS}$ whose premises occur earlier in σ. However, proofs can also be represented as prooftrees whose leaves all carry axioms and where lower nodes carry formulas that result by rule applications from formulas occurring immediately above. In this way the derivations in $\mathcal{FS}$ can inductively be defined as follows: every axiom of $\mathcal{FS}$ is a derivation in $\mathcal{FS}$. And if $\mathcal{D}_1, \ldots, \mathcal{D}_n$ are derivations in $\mathcal{FS}$ with respective conclusions $A_1, \ldots, A_n$, and if (4.1) is an application of a rule R of $\mathcal{FS}$, then the prooftree

$$\frac{\begin{array}{ccc} \mathcal{D}_1 & & \mathcal{D}_n \\ A_1 & \ldots & A_n \end{array}}{B}\, R$$

is a derivation in $\mathcal{FS}$ with conclusion B. We write $\vdash_{\mathcal{FS}} E$ and say that E is a theorem of $\mathcal{FS}$ if and only if there is a derivation $\mathcal{D}$ of $\mathcal{FS}$ with conclusion E.

For some purposes, like for investigating the question what kind of rules can be ‘modelled’ by using the axioms and rules of a formal system, one can be interested in what formulas are derivable in a formal system $\mathcal{FS}$ from some given assumptions, i.e. from given formulas of $\mathcal{FS}$. For this usually a consequence relation $\vdash_{\mathcal{FS}}$ between sets of formulas of $\mathcal{FS}$ (the *assumptions*) and formulas of $\mathcal{FS}$ (the respective *logical consequences*) is defined: for every formal system $\mathcal{FS}$ and formulas $A_1, \ldots, A_n, B$ of $\mathcal{FS}$, it is stipulated:

$$\{A_1, \ldots, A_n\} \vdash_{\mathcal{FS}} B \quad \Longleftrightarrow_{\text{def}} \quad \vdash_{\mathcal{FS}+\{A_1, \ldots, A_n\}} B \, . \tag{4.2}$$

The possibility of representing a consequence relation by using a formal system is called the *extension method* by Avron in [Avr91, p.24,25]: in (4.2) the consequence relation $\vdash_{\mathcal{FS}}$ is defined in terms of the provability relations $\vdash_{\mathcal{FS}+\Sigma}$ of the usually infinitely many extensions $\mathcal{FS}+\Sigma$ of $\mathcal{FS}$ that result by adding the formulas of Σ as new axioms to the axioms of $\mathcal{FS}$.

However, the use of infinitely many axiomatic systems for the sake of the definition of a consequence relation may be looked upon as conceptually unelegant and therefore be undesirable. It can be avoided by the use of the concept of "Hilbert system" in which derivations may start from unproven assumptions. The particular class of "pure Hilbert systems" is considered in the Subsection 4.2.3.

4.2.2 Local (and Not Local) Rules

Pure Hilbert-system rules are "local rules", in the sense of rules for "LR-systems" introduced by Troelstra and Schwichtenberg in [TS00], with the specific property that they have formulas as their "deduction elements". An *n-premise local rule* R is, following a definition in [TS00, p.77–78], a set of sequences $\langle \mathcal{S}_1, \ldots, \mathcal{S}_{n+1} \rangle$ of length $n+1$, where $\mathcal{S}_i$, $\mathcal{S}$ are *deduction elements* (like formulas or sequents) such that

$$\frac{\mathcal{S}_1 \ \ \ldots \ \ \mathcal{S}_n}{\mathcal{S}} \, R \tag{4.3}$$

is an application[1] of R whenever $\langle \mathcal{S}_1, \ldots, \mathcal{S}_n, \mathcal{S} \rangle \in R$, with *premises* $\mathcal{S}_1$, ..., $\mathcal{S}_n$ and with conclusion $\mathcal{S}$. [TS00] go on to define "LR-systems" as systems with a finite set of local rules, and with obvious definitions of "deduction tree", or "prooftree".

As an explanation for the term "local rules" [TS00] write that such rules

> "... are local in the sense that the correctness of a rule-application at a node ν can be decided locally, namely by looking at the name of the rule assigned to ν, and the proof-objects assigned to ν and its immediate successors (i.e. the nodes immediately above it)" ([TS00, p.76]).

Quite obviously, rules in natural-deduction systems that enable applications at which assumptions can be discharged do not conform to this format of rules; there, for the correctness of a rule application also the open assumptions present in respective subderivations have to be looked up, and if assumptions are discharged, it has to be made sure that this is indicated at the rule application in the right way. However, as [TS00] point out, natural-deduction systems can be brought under

[1] The rule name label for R is not drawn in a similar inference in [TS00, p.75], but it is supposed to be present in rule applications that appear in derivations (see the "Remarks" on p. 76 in [TS00]).

the definition of “LR-systems” by formulating them as corresponding sequent-style calculi (for example, see Section 2.1.8 in [TS00, p.41,42]).

Another form of non-local rules is able to appear in Hilbert systems in which derivations may contain assumptions: rules that do not discharge assumptions (as some rules in natural-deduction systems do), but that take the presence, or the absence, of assumptions in subderivations into account. A typical example of an *impure* Hilbert-system rule is the rule UG (for “universal generalization”) with applications of the form

$$\frac{\begin{array}{c}\mathcal{D}_1\\ A\end{array}}{\Box A}\,\mathrm{UG} \quad (\text{if } \mathcal{D} \text{ does not contain assumptions})\ . \tag{4.4}$$

This rule is used in standard Hilbert-system representations of the *truth* consequence relation in a normal modal logic (the letter A in (4.4) stands for an arbitrary formula in the respective modal logic). In systems for this purpose, an inference of a formula A from a formula $\Box A$ is clearly undesirable if A is *not true* in the respective normal modal logic; respective applications have therefore been excluded in the definition of UG in (4.4). Another example of an impure Hilbert-system rule will be given below in Example 4.2.3.

It is not difficult to formally describe impure Hilbert-system rules and (impure) Hilbert systems in a similar way as we treated pure Hilbert systems in Subsection 4.2.1: in contrast with pure Hilbert-system rules, for every application ι of an *impure* Hilbert-system rule next to the sequence of premises of ι and the conclusion of ι also the multiset of assumptions that are present for ι has to be specified. Furthermore, it is easy to adapt the definition of derivations in pure Hilbert systems (given in the next section) to impure Hilbert systems. However, formalizations of impure Hilbert systems can also be viewed as a subclass of natural-deduction-style systems, which we will consider in the Section 4.3.

4.2.3 Pure Hilbert Systems

By a *pure Hilbert system* we will mean a formal system that is endowed with a notion of derivation that may start from unproven assumptions. More precisely, we consider a pure Hilbert system $\mathcal{H}$ to be a 6-tuple $\langle Fo, Ax, \mathcal{R}, Der, \mathsf{assm}, \mathsf{concl}\rangle$ where

- Fo, Ax, and $\mathcal{R}$ are sets consisting of the *formulas*, the *axioms*, and the *rules* of $\mathcal{H}$,
- $Ax \subseteq Fo$ holds, i.e. every axiom of $\mathcal{H}$ is a formula of $\mathcal{H}$,
- every rule $R \in \mathcal{R}$ is a pure Hilbert-system rule with premises and conclusions in Fo,
- Der is a set, called the set of *derivations of* $\mathcal{H}$, that is inductively defined from Fo, Ax and $\mathcal{R}$ as follows, together with the functions $\mathsf{assm} : Der \to \mathcal{M}_{\mathrm{f}}(Fo)$

and $\mathsf{concl} : \mathit{Der} \to \mathit{Fo}$, which respectively assign the multiset $\mathsf{assm}(\mathcal{D})$ of *assumptions of* $\mathcal{D}$, and the *conclusion* $\mathsf{concl}(\mathcal{D})$ *of* $\mathcal{D}$ to every derivation $\mathcal{D} \in \mathit{Der}$:

Every axiom $A \in \mathit{Ax}$ is a derivation $\mathcal{D}$ of $\mathcal{H}$ with conclusion $\mathsf{concl}(\mathcal{D}) = A$ and without assumptions, i.e. with $\mathsf{assm}(\mathcal{D}) = \emptyset$. Every formula $A \in \mathit{Fo}$ is a derivation of $\mathcal{H}$ with $\mathsf{concl}(\mathcal{D}) = A$ and $\mathsf{assm}(\mathcal{D}) = \mathrm{mset}(\{A\})$. And furthermore, given formulas $A_1, \ldots, A_n, B \in \mathit{Fo}$ and derivations $\mathcal{D}_1, \ldots, \mathcal{D}_n$ of $\mathcal{H}$ with conclusions $\mathsf{concl}(\mathcal{D}_i) = A_i$ for all $1 \leq i \leq n$, and given that (4.1) is an application of a rule R of $\mathcal{H}$, then

$$\frac{\begin{matrix} \mathcal{D}_1 & & \mathcal{D}_n \\ A_1 & \ldots & A_n \end{matrix}}{B}\, R \tag{4.5}$$

is a derivation $\mathcal{D}$ in $\mathcal{H}$ with $\mathsf{concl}(\mathcal{D}) = B$ and $\mathsf{assm}(\mathcal{D}) = \biguplus_{i=1}^{n} \mathsf{assm}(\mathcal{D}_i)$.

For every pure Hilbert system $\mathcal{H}$, we will allow to refer to the sets of formulas, axioms, rules and derivations of $\mathcal{H}$ by $\mathit{Fo}_{\mathcal{H}}$, $\mathit{Ax}_{\mathcal{H}}$, $\mathcal{R}_{\mathcal{H}}$ and $\mathcal{D}\mathit{er}(\mathcal{H})$, respectively.

For defining the notions of “theorem”, “theory”, and (usual) “consequence relation” in a pure Hilbert system, let $\mathcal{H}$ be an arbitrary such system with set Fo of formulas. A formula of $\mathcal{H}$ is a *theorem* of $\mathcal{H}$ (notation $\vdash_{\mathcal{H}} A$) if and only if there is a derivation of $\mathcal{H}$ without assumptions and with conclusion A; more formally we set, for all formulas A of $\mathcal{H}$,

$$\vdash_{\mathcal{H}} A \quad \Longleftrightarrow_{\mathrm{def}} \quad (\exists\, \mathcal{D} \in \mathcal{D}\mathit{er}(\mathcal{H})) \left[\mathsf{assm}(\mathcal{D}) = \emptyset \;\;\&\;\; \mathsf{concl}(\mathcal{D}) = A \right] .$$

The *theory* $\mathit{Th}(\mathcal{H})$ of $\mathcal{H}$ is the set of theorems of $\mathcal{H}$. The ‘usual’ consequence relation $\vdash_{\mathcal{H}}$ on $\mathcal{H}$ is defined via the existence of derivations in $\mathcal{H}$ as follows: $\vdash_{\mathcal{H}} \subseteq \mathcal{P}(\mathit{Fo}) \times \mathit{Fo}$ is a relation between sets of formulas and formulas of $\mathcal{H}$ that is defined, for all sets $\Sigma \subseteq \mathit{Fo}$ and all formulas $A \in \mathit{Fo}$, by

$$\begin{aligned} \Sigma \vdash_{\mathcal{H}} A \quad \Longleftrightarrow_{\mathrm{def}} \quad (\exists\, \mathcal{D} \in \mathcal{D}\mathit{er}(\mathcal{H})) \big[& \mathrm{set}(\mathsf{assm}(\mathcal{D})) \subseteq \Sigma \;\;\& \\ & \&\;\; \mathsf{concl}(\mathcal{D}) = A \big] . \end{aligned} \tag{4.6}$$

Other consequence relations could be considered as well: for instance, one might for some special purpose be interested in the following ‘linear logic variant’ of the usual consequence relation in which a formula a is considered to be derivable from a multiset Γ of assumptions only if there is a derivation with conclusion A that uses arbitrary formulas precisely as often as they occur in $\mathcal{S}$. More precisely, for every pure Hilbert system $\mathcal{H}$, the relation $\vdash_{\mathcal{H}}^{(\mathrm{m})} \subseteq \mathcal{M}_{\mathrm{f}}(\mathit{Fo}) \times \mathit{Fo}$ between finite multisets of formulas and formulas of $\mathcal{H}$ is defined, for all finite sets $\Sigma \subseteq \mathit{Fo}$ and all formulas $A \in \mathit{Fo}$, by

$$\Gamma \vdash_{\mathcal{H}}^{(\mathrm{m})} A \quad \Longleftrightarrow_{\mathrm{def}} \quad (\exists\, \mathcal{D} \in \mathcal{D}\mathit{er}(\mathcal{H})) \left[\mathsf{assm}(\mathcal{D}) = \Gamma \;\;\&\;\; \mathsf{concl}(\mathcal{D}) = A \right] .$$

We will not consider this ‘non-standard’ consequence relation here (but we refer to [Gra03a], where also a variant notion of rule derivability is studied that is based on $\vdash^{(\mathrm{m})}$).

In connection with the consequence relation $\vdash_{\mathcal{H}}$ on a pure Hilbert system $\mathcal{H}$, we define also a notion of "mimicking derivation" between derivations in (possibly different) pure Hilbert systems. Hereby a derivation $\mathcal{D}_1$ is understood to "mimic" a derivation $\mathcal{D}_2$ if $\mathcal{D}_1$ has a more general 'input/output-behaviour' than $\mathcal{D}_2$ (here, for once, we consider the multiset of assumptions as the 'input' of a derivation and the conclusion as its 'output'). More formally, we denote, for all pure Hilbert systems $\mathcal{H}_1$ and $\mathcal{H}_2$ and derivations $\mathcal{D}_1 \in \mathcal{D}er(\mathcal{H}_1)$ and $\mathcal{D}_2 \in \mathcal{D}er(\mathcal{H}_2)$, by $\mathcal{D}_1 \precsim \mathcal{D}_2$ the assertion "$\mathcal{D}_1$ *mimics* $\mathcal{D}_2$" that is defined by

$$\mathcal{D}_1 \precsim \mathcal{D}_2 \iff_{\text{def}} \text{set}(\mathsf{assm}(\mathcal{D}_1)) \subseteq \text{set}(\mathsf{assm}(\mathcal{D}_2)) \;\; \& \\ \& \;\; \mathsf{concl}(\mathcal{D}_1) = \mathsf{concl}(\mathcal{D}_2)\,. \quad (4.7)$$

The relationship between the consequence relation $\vdash_{\mathcal{H}}$ of a pure Hilbert system $\mathcal{H}$ and the mimicking relation $\precsim$ between derivations in pure Hilbert systems is as follows: for every pure Hilbert system $\mathcal{H}$ and all formulas $A_1, \ldots, A_n, B \in Fo_{\mathcal{H}}$ it holds that

$$\{A_1, \ldots, A_n\} \vdash_{\mathcal{H}} B \iff (\exists \mathcal{D} \in \mathcal{D}er(\mathcal{H}))\left[\mathcal{D} \precsim \frac{A_1 \quad \ldots \quad A_n}{B}\right]. \quad (4.8)$$

It is easy to define a different "mimicking relation" $\simeq^{(\mathrm{m})}$ that allows to characterize the consequence relation $\vdash^{(\mathrm{m})}_{\mathcal{H}}$ on a pure Hilbert system $\mathcal{H}$ in an analogous way.

For the definition of rule derivability and admissibility below, we will use the following notations for extensions of pure Hilbert systems: For all pure Hilbert systems $\mathcal{H}$ with set Fo of formulas, for all sets $\Sigma \subseteq Fo$, and for all rules R on Fo we let

$$\mathcal{H}{+}R =_{\text{def}} \text{extension of } \mathcal{H} \text{ by adding the rule } R$$
$$\mathcal{H}{+}\Sigma =_{\text{def}} \text{extension of } \mathcal{H} \text{ by adding the formulas of } \Sigma \text{ as axioms}\,.$$

And furthermore, for all pure Hilbert systems $\mathcal{H}_1$ and $\mathcal{H}_2$, we denote by $\mathcal{H}_1 \sim_{th} \mathcal{H}_2$ the assertion "$\mathcal{H}_1$ and $\mathcal{H}_2$ are *equivalent*" and let

$$\mathcal{H}_1 \sim_{th} \mathcal{H}_2 \iff_{\text{def}} \mathcal{H}_1 \text{ and } \mathcal{H}_2 \text{ have the same theorems}\,.$$

4.2.4 Definitions of Rule Correctness, Admissibility and Derivability

We can now give the definition of rule admissibility and derivability in pure Hilbert systems. And we also give the definition for a notion of "rule correctness".

Definition 4.2.1. (Correctness, admissibility and derivability of rules in pure Hilbert systems). Let $\mathcal{H}$ be a pure Hilbert system and let R be a pure rule on the formulas of $\mathcal{H}$.

(i) The rule R is *correct for* $\mathcal{H}$ (R is a *correct rule for* $\mathcal{H}$) if and only if the collection of theorems of $\mathcal{S}$ is closed under applications of R, i.e. iff

$$\frac{A_1 \quad \ldots \quad A_n}{B} \;\text{ is an application of } R \quad \Longrightarrow$$

$$\Longrightarrow \quad \big[(\vdash_{\mathcal{H}} A_1) \;\&\; \ldots \;\&\; (\vdash_{\mathcal{H}} A_n) \;\Longrightarrow\; (\vdash_{\mathcal{H}} B) \big] \tag{4.9}$$

holds for all $n \in \omega$ and for all formulas $A_1, \ldots, A_n, B$ of $\mathcal{H}$.

(ii) The rule R is *admissible in* $\mathcal{H}$ (R is an *admissible rule of* $\mathcal{H}$) if and only

$$\mathcal{H}{+}R \sim_{th} \mathcal{H} \tag{4.10}$$

holds, i.e. iff extending $\mathcal{H}$ with the additional rule R does not lead to more theorems in the extended system $\mathcal{H}{+}R$.

(iii) The rule R is *derivable in* $\mathcal{H}$ (R is a *derivable rule of* $\mathcal{H}$) if and only if every application of R can be mimicked by a derivation in $\mathcal{H}$, i.e. iff

$$\frac{A_1 \quad \ldots \quad A_n}{B} \;\text{ is an application of } R \quad \Longrightarrow$$

$$\Longrightarrow \quad (\exists \mathcal{D} \in \mathcal{D}er(\mathcal{H}))\big[\, \mathcal{D} \;\precsim\; \frac{A_1 \quad \ldots \quad A_n}{B} \,\big] \,. \tag{4.11}$$

holds for all $n \in \omega$ and for all all formulas $A_1, \ldots, A_n, B$ of $\mathcal{H}$.

⊠

Remark 4.2.2. In the items (a)–(c) below we give, for each of the three notions defined in Definition 4.2.1, references to corresponding notions in the literature. In (a) and (b) we also remark on the reasons for distinguishing between "rule correctness" and "rule admissibility". And in (d) we explain that rule correctness is implied by rule admissibility.

(a) The stipulation in the above definition for "the rule R is correct for $\mathcal{H}$", where $\mathcal{H}$ is a pure Hilbert system, as "the theory of $\mathcal{H}$ is closed under applications of R" follows the use of the term "correct rule" by Scott in [Sco74, p.151] and corresponds to the definition of "dependent rule" ("abhängige Schlußregel") by Schmidt in [Schm60, p.149] as well as to the definition of "admissible rule" by Hindley/Seldin in [HS86, p.70] and Troelstra/Schwichtenberg in [TS00, p.76].

Although there is, as we will see in Lemma 4.2.4 below, a convincing reason for why conditions like (4.9) in Definition 4.2.1 are frequently taken as defining clauses for "rule admissibility" in pure Hilbert systems (or similar kinds of systems), we have decided not to follow this practice here, for three reasons. Firstly, the condition (4.9) on a rule R in a pure Hilbert system $\mathcal{H}$ is not in itself adequately reflected by calling R "admissible in $\mathcal{H}$" in case that (4.9) holds; fulfilledness of (4.9) is better described by saying that R "is correct for $\mathcal{H}$" (thus by following the terminology of Scott in [Sco74]). Secondly, the condition (4.9) is linked to the defining clause (4.10) of the more literal stipulation for rule admissibility only by an additional argument (a proof of Lemma 4.2.4, (i), below), however easy it is to provide such an argument. And thirdly, the clause (4.9) cannot be transferred in a straightforward manner to other kind of formal systems, such as natural-deduction style proof systems (this is different for the condition (4.10)).

It can easily be verified that rule correctness could alternatively be defined in the following way by using the notion of "mimicking derivation": R is correct for $\mathcal{H}$ if and only if every application of R that has only theorems of $\mathcal{H}$ as premises can be mimicked in $\mathcal{H}$, i.e. iff for all $n \in \omega$ and for all formulas $A_1, \dots, A_n, B$ of $\mathcal{H}$ it holds that

$$\frac{A_1 \;\dots\; A_n}{B} \;\text{ is an application of } R \quad\Longrightarrow$$
$$\Longrightarrow\ \Big[(\vdash_{\mathcal{H}} A_1) \;\&\; \dots \;\&\; (\vdash_{\mathcal{H}} A_n) \;\Rightarrow$$
$$\Rightarrow\ (\exists\, \mathcal{D}\in\mathcal{D}er(\mathcal{H}))\,[\,\mathcal{D} \precsim \frac{A_1 \;\dots\; A_n}{B}\,]\Big]\,.$$

(b) The stipulation in Definition 4.2.1, (ii), for an assertion "the rule R is admissible in $\mathcal{H}$", where $\mathcal{H}$ is a pure Hilbert system, as "the admission of R as an additional rule to the axioms and rules of the system $\mathcal{H}$ does not make it possible to prove more theorems" follows the definition of this notion by Lorenzen in [Lor69, p.19] [which, to my knowledge is an original definition, C.G.] and by Schütte in [Schu60, p.40] (who refers to Lorenzen); both Lorenzen and Schütte use the German expression "zulässige (Schluß-)Regel". It is also used by Curry in [Cur63, p.97], as well as in many more recent publications including, for example, by Iemhoff in [Iem01]. Admissible rules in this sense are called "derived" by Kleene in [Klee52, p.86].

In Definition 4.2.1, (ii), we have adopted the original definition of rule admissibility due to Lorenzen, for mainly two reasons: firstly, the condition imposed by (4.10) on a rule R in an APHS $\mathcal{H}$ is succinctly described by the expression "R is admissible in $\mathcal{H}$", and secondly, this definition lends itself immediately to being transferred from pure Hilbert systems to other kinds of proof systems such as natural-deduction systems (contrary to the clause (4.9) for rule correctness).

(c) The definition of rule derivability in pure Hilbert systems follows common definitions of this notion by, for instance, Kleene [Klee52, p.94] ("derived rule of *direct* type") Lorenzen [Lor69, p.26] ("Deduktionsprinzip"), Schütte [Schu60, p.42] ("'direkte Ableitbarkeit' von Schlüssen und Schlußregeln"), Curry [Cur63, p.97] (sentences that are "formally deducible" from other sentences), Scott [Sco74, p.153] (derivability of a rule from other rules), Hindley and Seldin [HS86, p.70], and Troelstra and Schwichtenberg [TS00, p.97].

We have drawn, in particular, from the definition of rule derivability by Hindley and Seldin in [HS86, p.70]. There, this notion is introduced in terms of the 'usual' consequence relation that, for the respective system, is defined as in (4.6). Indeed, it follows from the definition of the mimicking relation $\precsim$ that Definition 4.2.1, (iii), can be restated as follows: in every pure Hilbert systems $\mathcal{H}$ and for all pure Hilbert-style rules R on the formulas of *Fo*, R is derivable in $\mathcal{H}$ if and only if

$$\frac{A_1 \ \ldots \ A_n}{B} \ \text{ is an application of } R \quad\Longrightarrow\quad A_1, \ldots, A_n \vdash_{\mathcal{H}} B$$

holds for all $n \in \omega$ and all formulas $A_1, \ldots, A_n, B$ of $\mathcal{H}$. We have chosen, however, to define rule derivability in terms of the notion of "mimicking derivation", which explains rule derivability in a somewhat more explicit way.

(d) Rule correctness is a formally weaker notion than rule admissibility, and furthermore it is easier to show in general; in fact it can be looked upon as a *criterion* for proving rule admissibility (cf. Remark 4.2.4, (a), below).

To see that correctness is implied by admissibility, let $\mathcal{H}$ be a pure Hilbert system and let R be a pure Hilbert-system rule on a set *Fo* of formulas. "R is correct for $\mathcal{H}$" means that, *for all derivations* $\mathcal{D}$ in $\mathcal{H}{+}R$ *of the* particular *form* (4.5) (with only a single application of R, the one at the bottom), where $\mathcal{D}_1, \ldots, \mathcal{D}_n$ are derivations in $\mathcal{H}$ without assumptions, there exists a derivation $\mathcal{D}'$ in $\mathcal{H}$ without assumptions and with the same assumptions as $\mathcal{D}$. Contrasting with this, "R is admissible in $\mathcal{H}$" expresses the more general statement that *for all derivations* $\mathcal{D}$ in $\mathcal{H}{+}R$ without assumptions there exists a derivation $\mathcal{D}'$ in $\mathcal{H}$ without assumptions and with the same conclusion as $\mathcal{D}$. Hence, if R is admissible in $\mathcal{H}$, then R is also correct for $\mathcal{H}$. This argument shows the first sentence in Lemma 4.2.4, (ii), below.

Put more informally, correctness of a rule R with respect to a pure Hilbert system $\mathcal{H}$ means that *the* application of R can be eliminated from every derivation in $\mathcal{H}{+}R$ without assumptions and with only a single application of R, whereas admissibility of R with respect to $\mathcal{H}$ means the formally stronger assertion that *all* applications of R can be eliminated from every derivation in $\mathcal{H}{+}R$ without assumptions (and with an arbitrary number of R-applications).

Example 4.2.3. Let $\mathcal{H}$ be an arbitrary pure Hilbert system that completely axiomatizes the theory **C** of classical logic; the argument we give here can be transferred directly to pure Hilbert systems that completely axiomatize either of the theories **I** or **M** of intuitionistic or minimal predicate logic (see [TS00, p.35–38] for definitions of **M**, **I**, and **C** using respective natural-deduction systems **Nm**, **Ni**, and **Nc**).

The unrestricted generalization rule G^- with applications of the form

$$\frac{A[y/x]}{\forall x\, A}\,\mathrm{G}^- \quad (\text{if } y \equiv x \text{ or } y \notin \mathrm{fv}(A)) \tag{4.12}$$

is a pure Hilbert-system rule that is correct for $\mathcal{H}$. Indeed, whenever, for a formula A and variables x and y with $y \equiv x$ or $y \notin \mathrm{fv}(A)$, the formula $A[y/x]$ is a theorem of

$\mathcal{H}$, and hence of **C**, then $\forall x A$ is a theorem of **C**, and hence of $\mathcal{H}$. As a consequence of Proposition 4.2.4 below it follows that G^{-} is also admissible in $\mathcal{H}$.

From the rule G^{-} we have to distinguish the *generalization rule* G with applications of the form

$$\frac{\begin{array}{c}\mathcal{D}_1\\ A[x/y]\end{array}}{\forall x\, A}\,\mathrm{G} \quad \begin{array}{l}(\text{if } y \equiv x \text{ or } y \notin \mathrm{fv}(A), \text{ and } y \notin \mathrm{fv}(B)\\ \quad\text{for any assumption } B \text{ in } \mathcal{D}_1),\end{array} \tag{4.13}$$

which is *not* a rule for a pure Hilbert system because its applications are 'sensitive' to the presence of assumptions in subdervations (it is an *impure* Hilbert-system rule). G often appears in (impure) Hilbert-system axiomatizations of the consequence relation $\vdash_{\mathbf{C}}$ on formulas of **C** for which the deduction theorem

$$A \vdash_{\mathbf{C}} B \quad \Longleftrightarrow \quad \vdash_{\mathbf{C}} A \rightarrow B$$

holds, or in similar Hilbert-system axiomatizations for consequence relations on **I** and **M** with an analogous property. Examples of such (impure) Hilbert systems are the systems **Hc**, **Hi**, and **Hm** in [TS00, p.52] (in these systems the rule G is designated by ∀I, short for "for-all introduction", under which name it is familiar from natural-deduction systems for **C**, **I**, and **M**). ■

4.2.5 Basic Results

The following proposition, which is just an easy a reformulation of Lemma 6.14 in [HS86, p.70], gathers the most basic interconnections between the notions or rule correctness, admissibility and derivability in pure Hilbert systems. In its formulation we use the term "extension by enlargement" [2] : for all pure Hilbert systems $\mathcal{H}_1$, an *extension by enlargement* of $\mathcal{H}_1$ is a pure Hilbert system $\mathcal{H}_2$ that results from $\mathcal{H}_1$ by adding additional formulas, new axioms and/or new rules.

Lemma 4.2.4. *Let $\mathcal{H}$ be a pure Hilbert-system and let R be a pure Hilbert-system rule on the set of formulas of $\mathcal{H}$. Then the following statements holds:*

(i) R is correct for $\mathcal{H}$ $\Longleftrightarrow$ R is admissible in $\mathcal{H}$.

(ii) If R is derivable in $\mathcal{H}$, then R is also admissible in $\mathcal{H}$. The implication in the opposite direction does not hold in general.

(iii) If R is derivable in $\mathcal{H}$, then R is derivable in every extension by enlargement of $\mathcal{H}$.

A version of this lemma with respect to the precise notion of "abstract pure Hilbert system" (APHS), Lemma 4.2.4, is given in Appendix B.

[2] For a motivation of this term, see the paragraph before Definition B.1.6, starting on p. 365, in Appendix B.

Remark 4.2.5. The implication "$\Leftarrow$" in assertion (i) of Proposition 4.2.4 is obvious to see (cf. Remark 4.2.2 (d)), whereas proving the implication "$\Rightarrow$" involves an (easy) argument of successively eliminating all applications of R from derivations in $\mathcal{H}+R$ that do not contain assumptions. Hence rule correctness is indeed, as mentioned in Remark 4.2.2, a *valid* criterion for rule admissibility. A proof of "$\Rightarrow$" is sketched in [Schm60, p.150], and in [Cur63, p.97], while in many other expositions it is considered as a trivial matter. For instance, Schütte "hides" a proof for an assertion corresponding to Proposition 4.2.4, (i), between two sentences[3] in [Schu60, p.40]. As mentioned in Remark 4.2.2, (a), Hindley/Seldin in [HS86], and Troelstra / / Schwichtenberg in [TS00] use the term "admissible rule" for rules that are here called "correct rules" (respectively in relation to some APHS). Proposition 4.2.4, (i), can be viewed as a justification for this use of the term "admissible rule" since it entails

$$R \text{ is ``admissible'' in } \mathcal{H} \text{ (in the sense of [HS86], [TS00])} \quad \Longleftrightarrow \quad \mathcal{H}+R \sim_{th} \mathcal{H}$$

(which now is just a reformulation of Lemma 6.16, (i), in [HS86, p.70] in terms of our notation for equivalent systems and of adding new rules to pure Hilbert systems) and hence it states that the definitions of rule admissibility in [HS86] and in [TS00] coincide with the definition of rule admissibility in the more literal sense stipulated here (following [Lor69] and [Schu60]) *in pure Hilbert systems*. The fact that the statement "rule admissibility = rule correctness" (using these terms again according to Definition 4.2.1 again) *does not generalize* to natural-deduction systems *without complications* (as we will see in the next section) is the main reason why we have decided *not* to follow the definitions of "admissible rule" by Hindley and Seldin, and by Troelstra and Schwichtenberg.

The next theorem establishes a link between the assertions of items (ii) and (iii) in Proposition 4.2.4. It gives, for all pure Hilbert systems $\mathcal{H}$, two closely related characterizations of rule derivability in $\mathcal{H}$ in terms of rule admissibility in extensions of $\mathcal{H}$.

Theorem 4.2.6. *Let $\mathcal{H}$ be a pure Hilbert system with set Fo as its set of formulas, and let R be a pure Hilbert-system rule on Fo. Then the following three statements are equivalent:*

(i) R is derivable in $\mathcal{H}$.

(ii) R is admissible in every pure Hilbert system $\mathcal{H}+\Sigma$ with $\Sigma \in \mathcal{P}(Fo)$ arbitrary.

(iii) R is admissible in every extension by enlargement of $\mathcal{H}$.

[3] These two sentences in [Schu60, p.40] read as follows: "Eine *syntaktische Schlußregel* $\mathfrak{A}_1, \ldots, \mathfrak{A}_n \to \mathfrak{B}$, in der $\mathfrak{A}_1, \ldots, \mathfrak{A}_n, \mathfrak{B}$ Formelschemata eines formalen Systems Σ sind, heißt *zulässig* im System Σ, wenn die Hinzunahme dieser Schlußregel zu den Grundschlußregeln von Σ den Herleitbarkeitsbegriff des Systems Σ nicht ändert [emphasis as in the original, C.G.]. Die Schlußregel ist also genau dann zulässig, wenn sich in jedem Einzelfall der Formelschemata $\mathfrak{A}_1, \ldots, \mathfrak{A}_n, \mathfrak{B}$ aus der Herleitbarkeit der Prämissen auf die Herleitbarkeit der Konklusion schließen läßt."

This theorem is an informal version of Theorem 4.2.6, which is proved in Appendix B and which states an analogous relationship between rule derivability and admissibility in “abstract pure Hilbert systems”.

The following proposition contains two easy observations about the relationship between the notions of derivability and admissibility of an inference rule in an arbitrary pure Hilbert system $\mathcal{H}$. The first one is: if a rule R is admissible, but not derivable in $\mathcal{H}$, then there exists an application of R that contains at least one non-theorem as premise and that cannot be mimicked by a derivation in $\mathcal{H}$. And the related second observation is: for every admissible rule R in $\mathcal{H}$, the restriction of R to all those of its applications that only have theorems of $\mathcal{H}$ as premises is a derivable rule in $\mathcal{H}$.

Proposition 4.2.7. *Let $\mathcal{H}$ be a pure Hilbert system with set Fo of formulas, and let R be a pure Hilbert-system rule on Fo. Then the following two statements hold:*

(i) Suppose that R is admissible in $\mathcal{H}$. Then it holds that:

$$\begin{aligned} &R \textit{ is not derivable in } \mathcal{H} \iff \\ &\quad\iff (\exists\, n \in \omega)\, (\exists\, A_1, \ldots, A_n \in \textit{Fo}) \big[\text{(4.1) } \textit{is an application of } R \;\; \& \\ &\qquad\qquad \& \;\; \big((\nvdash_{\mathcal{H}} A_1) \vee \ldots \vee (\nvdash_{\mathcal{H}} A_n)\big) \;\; \& \;\; A_1, \ldots, A_n \nvdash_{\mathcal{H}} B \big] \;. \end{aligned}$$

(ii) Let R_0 be the rule that arises by restricting the applications of R to all those that exclusively have theorems of $\mathcal{H}$ as premises. Then it holds that

$$R \textit{ is admissible in } \mathcal{H} \quad\iff\quad R_0 \textit{ is derivable in } \mathcal{H} \;.$$

This proposition is an easy consequence of the definition of rule correctness in Definition 4.2.1, (i), and of Lemma 4.2.4, (i), the equivalence of the notions of rule correctness and rule admissibility in pure Hilbert systems.

4.2.6 Rule Elimination

In the following theorem the notions of rule admissibility and derivability with respect to a pure Hilbert system $\mathcal{H}$ are characterized in terms of from which derivations in $\mathcal{H}+R$ the applications of R can be eliminated. For the formulation of this theorem we stipulate the following: for all pure Hilbert systems $\mathcal{H}$, for all pure Hilbert-system rules R on the formulas of $\mathcal{H}$, and for all derivations $\mathcal{D}$ of $\mathcal{H}+R$, we say that the applications of R in $\mathcal{D}$ *can be eliminated* if and only if there exists a derivation $\mathcal{D}'$ of $\mathcal{H}$ that mimics $\mathcal{D}$.

Theorem 4.2.8. (Elimination of derivable and admissible rules). *Let $\mathcal{H}$ be a pure Hilbert system with set Fo of formulas, and let R be a pure Hilbert-system rule on Fo.*

(i) R is admissible in $\mathcal{H}$ if and only if R can be eliminated from all derivations $\mathcal{D}$ of $\mathcal{H}$ that do not contain assumptions. This means the following holds:

$$\begin{aligned} R \textit{ is admissible in } \mathcal{H} \iff & (\forall \mathcal{D} \in \mathcal{D}er(\mathcal{H}{+}R)) \left[\mathsf{assm}(\mathcal{D}) = \emptyset \implies \right. \\ & \left. \implies (\exists\, \mathcal{D}' \in \mathcal{D}er(\mathcal{H})) \left[\mathcal{D}' \precsim \mathcal{D} \right] \right] . \end{aligned}$$

(ii) R is derivable in $\mathcal{H}$ if and only if R can be eliminated from every derivation $\mathcal{D}$ of $\mathcal{H}$. More formally, the following holds:

$$R \textit{ is derivable in } \mathcal{H} \iff (\forall \mathcal{D} \in \mathcal{D}er(\mathcal{H}{+}R))\, (\exists\, \mathcal{D}' \in \mathcal{D}er(\mathcal{H})) \left[\mathcal{D}' \precsim \mathcal{D} \right] .$$

A version of this theorem with respect to the precise concept of "abstract pure Hilbert system" is Theorem B.1.15 in Appendix B.

4.3 Definitions and Results in Natural-Deduction Systems

Natural-deduction systems are due to Gentzen in [Gen35], who introduced the calculi NJ and NK for formalized 'natural reasoning' in intuitionistic and classical predicate logic. (However, in [TS00] it is pointed out that shortly prior to Gentzen a similar formalism—in linear, not in tree format—has already been introduced by Jaśkowski in 1934.) Later, thoroughly important work on natural-deduction systems concerning the concept of 'normalization' of derivations in such systems has been done by Prawitz in [Pra65], based on rigorous definitions of systems for intuitionistic predicate logic.

In Subsection 4.3.1 of this section we give a general description of natural-deduction systems in their 'usual', and that is, not sequent-style, formulations. Again, we do not base ourselves on completely formalized concepts of natural-deduction systems and of rules for such systems, but we only use rough descriptions of these systems instead; however, we frequently refer to precise formulations with respect to the concept of "abstract natural-deduction system" that is introduced in Section B.2 of Appendix B. The notation we use for natural-deduction systems is drawn mainly from the way how these systems are treated formally in [TS00].

In Subsection 4.3.2 we argue that the definitions of rule derivability, and in particular, of rule correctness cannot merely be taken over from the stipulations in pure Hilbert systems. Subsequently in Subsection 4.3.3, we introduce the notions of rule admissibility, "rule cr-correctness", "rule cr-admissibility", and rule derivability in natural-deduction systems. Eventually in Subsection 4.3.4, we give basic results concerning the relationships of the for introduced notions; lastly in Subsection 4.3.5 we give a result that relates the notions of admissibility and cr-admissibility with respective notions of rule elimination in natural-deduction systems.

4.3.1 Natural-Deduction Systems

Natural-deduction-style proof systems, here only called *natural-deduction systems*, are distinguished by the special feature that derivations may start from unproven assumptions which can be discharged only later at occurrences of appropriate rule applications. A derivation $\mathcal{D}$ in a natural-deduction system typically contains assumptions of which the conclusion of $\mathcal{D}$ has already been made independent, the "discharged" or "closed" assumptions of $\mathcal{D}$, and it may also contain assumptions on which the conclusion of $\mathcal{D}$ still depends, the "undischarged" or "open" assumptions of $\mathcal{D}$.

Let $\mathcal{S}$ be a natural-deduction system. We will denote by $Fo_{\mathcal{S}}$ the set of *formulas* of $\mathcal{S}$, by $Mk_{\mathcal{S}}$ the set of *assumption markers* of $\mathcal{S}$, and by

$$mFo_{\mathcal{S}} =_{\mathrm{def}} \{ A^u \mid u \in Mk_{\mathcal{S}} \}$$

the set of *marked formulas* of $\mathcal{S}$. Furthermore we let $\mathcal{D}er(\mathcal{S})$ be the *set of derivations* of $\mathcal{S}$. For every derivation $\mathcal{D} \in \mathcal{D}er(\mathcal{S})$ we will denote by $\mathsf{concl}(\mathcal{D})$ the *conclusion* of $\mathcal{D}$, and by $\mathsf{omassm}(\mathcal{D})$ the set of *open marked assumptions* of $\mathcal{D}$, i.e. the set of those assumptions of $\mathcal{D}$ that are not discharged in $\mathcal{D}$.

Let $\mathcal{S}$ again be a natural-deduction system. An application ι of a rule R of $\mathcal{S}$ is an inference that at the bottom of a derivation $\mathcal{D} \in \mathcal{D}er(\mathcal{S})$ has the following general form

$$\iota\ \dfrac{\begin{matrix}\{[C_i]^{u_i}\}_{i=1,\ldots,m} \\ \mathcal{D}_1 \\ A_1\end{matrix} \qquad \ldots \qquad \begin{matrix}\{[C_i]^{u_i}\}_{i=1,\ldots,m} \\ \mathcal{D}_n \\ A_n\end{matrix}}{B}\ R,\, u_1, \ldots, u_m \tag{4.14}$$

where, for some $n \in \omega$, $\mathcal{D}_1, \ldots, \mathcal{D}_n \in \mathcal{D}er(\mathcal{S})$ are the *immediate subderivations* of ι, $A_1, \ldots, A_n$ are the *premises* of ι, B is the *conclusion* of ι, and for some $m \in \omega$, the family $\{[C_i]^{u_i}\}_{i=1,\ldots,m}$ shown at the top of $\mathcal{D}_1, \ldots, \mathcal{D}_n$ gathers all those classes $[C_1]^{u_1}, \ldots, [C_m]^{u_m}$ of open marked assumptions with occurrences in one or more of $\mathcal{D}_1, \ldots, \mathcal{D}_n$ that are *discharged* at ι (the respective markers $u_1, \ldots, u_m$ of the open marked assumptions $C_1^{u_1}, \ldots, C_m^{u_m}$ that are discharged at ι are also attached to the inference ι). An occurrence of a marked assumption $D^v \in mFo_{\mathcal{S}}$ in $\mathcal{D}$ is called *open* or *undischarged* if and only if it corresponds to an open occurrence of D^v in one of the subderivations $\mathcal{D}_1, \ldots, \mathcal{D}_n$ of ι and if D^v is different from all marked assumptions $C_1^{u_1}, \ldots, C_m^{u_m}$; otherwise an occurrence of D^v is called *discharged* or *closed*. In particular this means that the following two statements hold about the relationship between the open marked assumptions $\mathsf{omassm}(\mathcal{D})$ of the derivation $\mathcal{D}$ in (4.14) ending with the application of ι and the open marked assumptions

$\mathsf{omassm}(\mathcal{D}_i)$ of the immediate subderivations $\mathcal{D}_1, \ldots, \mathcal{D}_n$ of ι:

$$\{C_1^{u_1}, \ldots, C_m^{u_m}\} \subseteq \bigcup_{i=1}^{n} \mathsf{omassm}(\mathcal{D}_i) \ , \tag{4.15}$$

$$\mathsf{omassm}(\mathcal{D}) = \Big(\bigcup_{i=1}^{n} \mathsf{omassm}(\mathcal{D}_i)\Big) \setminus \{C_1^{u_1}, \ldots, C_m^{u_m}\} \ . \tag{4.16}$$

Let $\mathcal{S}_1$ and $\mathcal{S}_2$ be natural-deduction systems. For all derivations $\mathcal{D}_1 \in \mathcal{D}er(\mathcal{S}_1)$ and $\mathcal{D}_2 \in \mathcal{D}er(\mathcal{S}_2)$, we denote by $\mathcal{D}_1 \simeq \mathcal{D}_2$ the statement "$\mathcal{D}_1$ *mimics* $\mathcal{D}_2$" that we stipulate to be true if and only if $\mathcal{D}_1$ and $\mathcal{D}_2$ have the same conclusion and the same open marked assumptions; more formally, we define

$$\mathcal{D}_1 \simeq \mathcal{D}_2 \iff_{\text{def}} \mathsf{omassm}(\mathcal{D}_1) = \mathsf{omassm}(\mathcal{D}_2) \ \& \ \mathsf{concl}(\mathcal{D}_1) = \mathsf{concl}(\mathcal{D}_2) \tag{4.17}$$

for all $\mathcal{D}_1 \in \mathcal{D}er(\mathcal{S}_1)$ and $\mathcal{D}_2 \in \mathcal{D}er(\mathcal{S}_2)$.

For all natural-deduction systems $\mathcal{S}$, we furthermore introduce the consequence relation $\vdash_{\mathcal{S}} \subseteq \mathcal{P}(mFo_{\mathcal{S}}) \times Fo_{\mathcal{S}}$ that is defined, for all $\tilde{\Sigma} \in \mathcal{P}(mFo_{\mathcal{S}})$ and $A \in Fo_{\mathcal{S}}$, by

$$\tilde{\Sigma} \vdash_{\mathcal{S}} A \iff_{\text{def}} (\exists \mathcal{D} \in \mathcal{D}er(\mathcal{S}))\big[\mathsf{omassm}(\mathcal{D}) = \tilde{\Sigma} \ \& \ \mathsf{concl}(\mathcal{D}) = A\big] \ . \tag{4.18}$$

The consequence relation $\vdash_{\mathcal{S}}$ is a stricter variant of a perhaps more frequently used consequence relation $\vdash'_{\mathcal{S}} \subseteq \mathcal{P}(Fo_{\mathcal{S}}) \times Fo_{\mathcal{S}}$ that, for all $\Sigma \in \mathcal{P}(Fo_{\mathcal{S}})$ and for all $A \in Fo_{\mathcal{S}}$ is defined as follows: $\Sigma \vdash'_{\mathcal{S}} A$ holds if and only if there is a derivation $\mathcal{D}$ in $\mathcal{S}$ with conclusion A and such that the formulas occurring in the marked assumptions $\mathsf{omassm}(\mathcal{D})$ of $\mathcal{D}$ are contained in Σ. As a reason for why we use $\vdash_{\mathcal{S}}$ here instead of $\vdash'_{\mathcal{S}}$, we want to hint that a notion of "cr-admissibility" (see Definition 4.3.2, (iii), below) would not be as 'well-behaved' if it were to be defined in terms of $\vdash'_{\mathcal{S}}$ instead of in terms of $\vdash_{\mathcal{S}}$ (except in so called "pure natural-deduction systems", see [Avr91, p.28,29]).

Similarly as for pure Hilbert systems, we introduce the notion "extension by enlargement": for all natural-deduction systems $\mathcal{S}_1$ and $\mathcal{S}_2$, we call $\mathcal{S}_2$ an *extension by enlargement* of $\mathcal{S}_1$ if and only if $\mathcal{S}_2$ results from $\mathcal{S}_1$ by adding new formulas, markers, and/or rules.

4.3.2 Problems with Naive Definitions of Rule Correctness and Rule Derivability

As a consequence of the more complex structure of inference rules in natural-deduction systems, the definitions of the notion of rule correctness and rule derivability cannot just be taken over from the stipulations for pure Hilbert systems in Definition 4.2.1. The stipulation for rule admissibility in Definition 4.2.1 (ii), however, will also be adopted for natural-deduction systems since it does not make explicit mention of the formal structure of rules and derivations in these systems.

Apart from that it is not immediately clear how the clauses (i) and (iii) of Definition 4.2.1 can be transferred to reach stipulations for rule correctness and derivability which apply to all possible rules in a natural-deduction system (and in particular to rules that allow assumptions to be discharged), there is yet another reason why some care has to be taken for adapting these notions in natural-deduction systems. Namely, incautious definitions would violate the desirable aim of preserving the most basic relationships between rule correctness, admissibility and derivability towards each other as known from pure Hilbert systems. This is because the stipulations in Definition 4.2.1 could very well provide meaningful definitions for rule correctness, admissibility and derivability for such rules in natural-deduction systems at which no assumptions are discharged. But the following example shows that, if these stipulations were adapted *naively*, then rule correctness would be a strictly weaker notion than rule admissibility, contrasting with the situation in pure Hilbert systems (see Proposition 4.2.4 (i)).

Example 4.3.1. We saw in Example 4.2.3 that the (pure Hilbert-system) rule G^- of unrestricted generalization (cf. (4.12)) is correct and admissible in every pure Hilbert system $\mathcal{H}$ for classical predicate calculus $\mathbf{C}$, and hence that adding G^- to such a system $\mathcal{H}$ does not lead to more theorems in the extended system. The situation changes, however, if the rule G^- is added, in the form of the *unrestricted* forall-introduction rule $(\forall\mathrm{I})^-$ with applications of the form

$$\begin{array}{c}\mathcal{D}_1\\ \dfrac{A[y/x]}{\forall x\,A}\,(\forall\mathrm{I})^-\end{array}\quad (\text{if } y\equiv x \text{ or } y\notin \mathrm{fv}(A))\,, \tag{4.19}$$

to a natural-deduction system for $\mathbf{C}$ like the system $\mathbf{Nc}$ given in [TS00, p.30]. Let us remark that the system $\mathbf{Nc}$ does actually contain the (restricted form of the) forall-introduction rule $\forall\mathrm{I}$ with applications of the form

$$\begin{array}{c}\mathcal{D}_1\\ \dfrac{A[x/y]}{\forall x\,A}\,\forall\mathrm{I}\end{array}\quad \begin{array}{l}(\text{if } y\equiv x \text{ or } y\notin \mathrm{fv}(A), \text{ and if } y\notin \mathrm{fv}(B)\\ \quad \textit{for all assumptions } B \textit{ that are open in } \mathcal{D}_1).\end{array} \tag{4.20}$$

However, if $(\forall\mathrm{I})^-$ is added to $\mathbf{Nc}$, then formulas become derivable in $\mathbf{Nc}$+$(\forall\mathrm{I})^-$ that are not theorems of $\mathbf{C}$:

$$\dfrac{\dfrac{\exists x A^{\,u} \qquad \dfrac{A^{v}}{\forall x A}\,(\forall\mathrm{I})^-}{\forall x A}\,\exists\mathrm{I}, v}{\exists x A \;\to\; \forall x A}\,\to\!\mathrm{I}, u \tag{4.21}$$

But on the other hand, the set of theorems of $\mathbf{Nc}$ (and hence of $\mathbf{C}$) is clearly still closed under applications of $(\forall\mathrm{I})^-$. Hence, if we were to base ourselves solely on the stipulations (i) and (ii) of Definition 4.2.1 (which indeed make sense for $(\forall\mathrm{I})^-$!), we would be tempted to call the rule $(\forall\mathrm{I})^-$ is correct for $\mathbf{Nc}$, while it is certaily not "admissible" in $\mathbf{Nc}$ (in the literal meaning of this term). ■

We conclude from this example that in natural-deduction systems rule correctness, if it were defined according to Definition 4.2.1, would no longer be a useful criterion for showing rule admissibility. A closer analysis of this phenomenon shows that correctness in the sense of Definition 4.2.1, (i), of an 'assumption-insensitive' rule R (like G^{-}) with respect to a natural-deduction system $\mathcal{S}$ only guarantees that no non-theorems of $\mathcal{S}$ are derivable by *such* derivations without open assumptions in $\mathcal{S}{+}R$ *that* do not contain applications of R having immediate subderivations in which open assumptions are present; other derivations in $\mathcal{S}{+}R$ without open assumptions may very well possess non-theorems of $\mathcal{S}$ as conclusions (as the derivation (4.21) in Example 4.3.1 demonstrates).

4.3.3 Definitions of Rule Cr-Correctness, Admissibility and Derivability

In the following definition, we propose some more careful definitions for rule correctness, admissibility and derivability in natural-deduction systems. More precisely, we introduce, for rules in a natural-deduction system, a notion of admissibility, notions of "cr-correctness" and "cr-admissibility" (i.e. correctness, and respectively admissibility, with respect to the $\underline{c}$onsequence $\underline{r}$elation in a system), and a notion of derivability.

For the definition of rule derivability, we need the notion of "derivation context" in a natural-deduction system, which we introduce now. Let $k \in \omega$. By a k-ary *derivation context* $\mathcal{DC}$ in a natural-deduction system $\mathcal{S}$ we understand the result of replacing within a derivation $\mathcal{D} \in \mathcal{D}er(\mathcal{S}')$, where $\mathcal{S}'$ is an extension by enlargement of $\mathcal{S}$, some subderivations by any of the holes $[\,]_1, \ldots, [\,]_k$ such that $\mathcal{DC}$ contains only rule applications of $\mathcal{S}$ (but not any more applications of rules of $\mathcal{S}'$ that are not also rules of $\mathcal{S}$); the set of all k-ary derivation contexts in $\mathcal{S}$ is denoted by $\mathcal{D}er\mathcal{C}txt_k(\mathcal{S})$. For a more precise definition of the notion of "derivation context" within the concept of "abstract natural-deduction system" by means of the notions "pseudo-derivation" and "pseudo-derivation context" in ANDS's, we refer to Definition B.2.13 in Section B.2 of Appendix B.

Definition 4.3.2. ((Cr-)Admissibility, correctness and derivability of rules in natural-deduction-style systems). Let $\mathcal{S}$ be a natural-deduction system with set $Fo_{\mathcal{S}}$ of formulas, and let R be a rule for a natural-deduction system with formulas $Fo_{\mathcal{S}}$ and assumption markers Mk.

(i) The rule R is *admissible* in $\mathcal{S}$ if and only if

$$\mathcal{S}{+}R \sim_{th} \mathcal{S} \tag{4.22}$$

holds, i.e. iff extending $\mathcal{S}$ with the additional rule R does not lead to more theorems in the extended system $\mathcal{S}{+}R$.

(ii) The rule R is *cr-correct* for $\mathcal{S}$ (R is *correct for $\mathcal{S}$ with respect to the $\underline{c}$onsequence $\underline{r}$elation* $\vdash_{\mathcal{S}}$) if and only if an application of R can always be eliminated from such derivations in $\mathcal{S}{+}R$ that contain an application of R at the bottom, but no other applications of R. More precisely, R is cr-correct for $\mathcal{S}$ iff, for every derivation $\mathcal{D} \in \mathcal{D}er(\mathcal{S}{+}R)$ of the form (4.14) such that

- $n, m \in \omega$,
- $C_1^{u_1}, \ldots, C_m^{u_m} \in mFo_{\mathcal{S}}$ are distinct marked formulas, and $A_1, \ldots, A_n, B \in Fo_{\mathcal{S}}$,
- the conditions (4.15) and (4.16) are fulfilled for the relationship between the open marked assumptions of $\mathcal{D}$ and those of its immediate subderivations $\mathcal{D}_1, \ldots, \mathcal{D}_n$, } (4.23)

- and $\mathcal{D}_1, \ldots, \mathcal{D}_n \in \mathcal{D}er(\mathcal{S})$, } (4.24)

holds, there exists a derivation $\mathcal{D}' \in \mathcal{D}er(\mathcal{S})$ such that $\mathcal{D}' \simeq \mathcal{D}$ is the case (i.e. such that $\mathcal{D}'$ mimics $\mathcal{D}$).

(iii) The rule R is *cr-admissible in* $\mathcal{S}$ (R is *admissible in* $\mathcal{S}$ *with respect to the consequence relation* $\vdash_{\mathcal{S}}$) if and only if

$$\vdash_{\mathcal{S}+R} \; = \; \vdash_{\mathcal{S}} \tag{4.25}$$

holds, i.e. iff the consequence relations $\vdash_{\mathcal{S}+R}$ on $\mathcal{S}{+}R$ and $\vdash_{\mathcal{S}}$ on $\mathcal{S}$ (both defined according to (4.18)) coincide.

(iv) The rule R is *derivable* in $\mathcal{S}$ if and only if applications of R can always be eliminated from such derivations of the form (4.14) in extensions by enlargement of $\mathcal{S}$ in the following special way: for every derivation $\mathcal{D}$ in an extension by enlargement $\mathcal{S}_{\text{ext}}$ of $\mathcal{S}$ containing R such that $\mathcal{D}$ is of the form (4.14), with (4.23) and $\mathcal{D}_1, \ldots, \mathcal{D}_n \in \mathcal{D}er(\mathcal{S}_{\text{ext}})$ fulfilled, there exists a derivation-context $\mathcal{DC}' \in \mathcal{D}er\mathcal{C}txt_n(\mathcal{S})$ such that the prooftree

$$\begin{array}{ccc}
\{[C_i]^{u_i}\}_{i=1,\ldots,m} & & \{[C_i]^{u_i}\}_{i=1,\ldots,m} \\
\mathcal{D}_1 & & \mathcal{D}_n \\
[A_1]_1 & \ldots & [A_n]_n \\
& \mathcal{DC}' & \\
& B &
\end{array} \tag{4.26}$$

(which is the result $\mathcal{DC}'[\mathcal{D}_1, \ldots, \mathcal{D}_n]$ of hole filling in the derivation-context $\mathcal{DC}$ with with $\mathcal{D}_1, \ldots, \mathcal{D}_n$) is a derivation $\mathcal{D}' \in \mathcal{D}er(\mathcal{S}_{\text{ext}})$ that mimics $\mathcal{D}$ (in particular this means that $\mathcal{D}'$ does not contain applications of R if R is not a rule of $\mathcal{S}$, and that all open marked assumptions $C_1^{u_1}, \ldots, C_m^{u_m}$ in $\mathcal{D}_1, \ldots, \mathcal{D}_n$ get discharged at rule applications in $\mathcal{D}'$ within the derivation-context $\mathcal{DC}'$).

⊠

A version of this definition with respect to the precise concept of "abstract natural-deduction system" (ANDS) is given in Definition B.2.21, Appendix B. There, cr-correctness of a rule R with respect to an ANDS $\mathcal{S}$ is defined equivalently in terms of the consequence relation $\vdash_{\mathcal{S}}$ as defined in (4.18); furthermore, derivability of a rule R in an ANDS $\mathcal{S}$ is defined in a different, but equivalent, way by relying on a concept of "pseudo-derivations" in an ANDS.

In Example 4.3.3 and Example 4.3.4 below, we illustrate the notion of rule derivability, for which the defining clause in Definition 4.3.2 is the most complicated one. We will later encounter examples of rules that are cr-admissible (and cr-correct as well as admissible) with respect to a natural-deduction system; in particular we refer to Proposition 6.2.7 for the case of three quite naturally appearing rules (in a proof system for recursive type equality) that are admissible, but not derivable with respect to a natural-deduction system and that have only applications at which assumptions get discharged.

Example 4.3.3 (Rule derivability in natural-deduction systems). For an example of rule derivability in a natural-deduction system let us consider the rule $\lor{\to}\lor\mathrm{I}$ that allows applications of the form

$$\dfrac{\begin{array}{cc} [A]^u & [B]^v \\ \mathcal{D}_1 & \mathcal{D}_2 \\ C & D \end{array}}{A \lor B \to C \lor D}\;\lor{\to}\lor\mathrm{I}\,, u, v \tag{4.27}$$

with respect to an arbitrary one of the natural-deduction systems $\mathbf{N}[\mathbf{mic}]$ for minimal, intuitionistic or classical (predicate) logic that are given in [TS00, p.30]. Among rules for other connectives, the systems $\mathbf{N}[\mathbf{mic}]$ contain the rules $\lor\mathrm{I}_{\mathrm{R}}$, $\lor\mathrm{I}_{\mathrm{L}}$, $\lor\mathrm{E}$, and $\to\mathrm{I}$ with applications of the form

$$\dfrac{\begin{array}{c} \mathcal{D}_1 \\ E \end{array}}{E \lor F}\;\lor\mathrm{I}_{\mathrm{R}} \qquad \dfrac{\begin{array}{c} \mathcal{D}_1 \\ F \end{array}}{E \lor F}\;\lor\mathrm{I}_{\mathrm{L}} \qquad \dfrac{\begin{array}{ccc} & [E]^u & [F]^v \\ \mathcal{D}_1 & \mathcal{D}_2 & \mathcal{D}_3 \\ E \lor F & G & G \end{array}}{G}\;\lor\mathrm{E}\,, u, v$$

$$\dfrac{\begin{array}{c} [E]^u \\ \mathcal{D}_1 \\ F \end{array}}{E \to F}\;\to\mathrm{I}, u$$

Let now $\mathcal{S}$ be an arbitrary one of the systems $\mathbf{N}[\mathbf{mic}]$. Then we find that the rule $(\lor{\to}\lor\mathrm{I})$ is derivable in $\mathcal{S}$. To recognize this, we observe that every derivation $\mathcal{D}$ of the form (4.27) in an extension by enlargement $\mathcal{S}_{\mathrm{ext}}$ of $\mathcal{S}$ containing $(\lor{\to}\lor\mathrm{I})$ (where hence also $\mathcal{D}_1$ and $\mathcal{D}_2$ are derivations in $\mathcal{S}_{\mathrm{ext}}$), can be mimicked by a derivation $\mathcal{D}' \in \mathcal{D}er(\mathcal{S})$ of the following form:

$$\dfrac{\dfrac{(A \lor B)^w \qquad \dfrac{\begin{array}{c} [A]^u \\ \mathcal{D}_1 \\ C \end{array}}{C \lor D}\;\lor\mathrm{I}_{\mathrm{R}} \qquad \dfrac{\begin{array}{c} [B]^v \\ \mathcal{D}_2 \\ D \end{array}}{C \lor D}\;\lor\mathrm{I}_{\mathrm{L}}}{C \lor D}\;\lor\mathrm{E}\,, u, v}{A \lor B \to C \lor D}\;\to\mathrm{I}\,, w$$

Hereby $\mathcal{D}'$ is clearly of the form $\mathcal{DC}'[\mathcal{D}_1, \mathcal{D}_2]$ for the following binary derivation-context $\mathcal{DC}' \in \mathcal{D}er\mathcal{C}txt_2(\mathcal{S})$:

$$\dfrac{\dfrac{(A \vee B)^w \qquad \dfrac{[\,]_1}{C \vee D}\,\vee\mathrm{I_R} \qquad \dfrac{[\,]_2}{C \vee D}\,\vee\mathrm{I_L}}{C \vee D}\,\vee\mathrm{E}\,, u, v}{A \vee B \to C \vee D}\,\to\mathrm{I}\,, w$$

■

Example 4.3.4 (Rule derivability in natural-deduction systems).

(i) We want to consider the rule TND (*"tertium non datur"*) that has only zero-premise applications of the form

$$\dfrac{}{A \vee \neg A}\,\mathrm{TND} \tag{4.28}$$

with respect to the natural-deduction system **Nc** for classical (predicate) logic that is defined in [TS00, p.30].

We find that TND is derivable in **Nc** by observing that every application of TND, and hence every derivation $\mathcal{D}$ of the form (4.28) can be mimicked by a derivation $\mathcal{D}' \in \mathcal{D}er(\mathbf{Nc})$ of the following form:

$$\dfrac{\dfrac{(\neg(A \vee \neg A))^u \qquad \dfrac{\dfrac{\dfrac{(\neg(A \vee \neg A))^u \qquad \dfrac{A^v}{A \vee \neg A}\,\vee\mathrm{I_R}}{\bot}\,\to\mathrm{E}}{\neg A}\,\to\mathrm{I},\ v}{A \vee \neg A}\,\vee\mathrm{I_L}}{\bot}\,\to\mathrm{E}}{A \vee \neg A}\,\bot_{\mathrm{c}},\ u$$

Obviously, $\mathcal{D}'$ is also a 0-ary derivation-context in $\mathcal{DC} \in \mathcal{D}er\mathcal{C}txt_0(\mathbf{Nc})$.

(ii) On the other hand, we want to consider the *classical absurdity rule* $\bot_c$ with applications of the form

$$\begin{array}{c}[\neg A]^u \\ \mathcal{D}_1 \\ \dfrac{\bot}{A}\,\bot_{\mathrm{c}},\ u\end{array} \tag{4.29}$$

with respect to the system **Ni**+TND. The system **Ni** contains the *intuitionistic absurdity rule* $\bot_{\mathrm{i}}$ that enables applications of the form

$$\begin{array}{c}\mathcal{D}_1 \\ \dfrac{\bot}{A}\,\bot_{\mathrm{i}}\end{array}$$

but it does not contain $\bot_{\mathrm{c}}$.

$\bot_{\mathrm{c}}$ is derivable in **Ni**+TND as a consequence of the fact that for every derivation $\mathcal{D}$ of the form (4.29) in an extension by enlargement $\mathcal{S}_{\mathrm{ext}}$ of **Ni**+$\bot_{\mathrm{c}}$ there exists a derivation $\mathcal{D}'$ of the form

$$\cfrac{\cfrac{}{A \lor \neg A}\,\text{TND} \qquad A^v \qquad \cfrac{\begin{matrix}[\neg A]^u \\ \mathcal{D}_1 \\ \bot\end{matrix}}{A}\,\bot_{\mathrm{i}}}{A}\,\lor\mathrm{E}, v, u$$

in $\mathcal{S}_{\mathrm{ext}}$, which is of the form $\mathcal{DC}'[\mathcal{D}_1]$ for the derivation-context $\mathcal{DC}'$ of the form

$$\cfrac{\cfrac{}{A \lor \neg A}\,\text{TND} \qquad A^v \qquad \cfrac{[\,]_1}{A}\,\bot_{\mathrm{i}}}{A}\,\lor\mathrm{E}, v, u$$

in **Ni**+TND, i.e. for which $\mathcal{DC} \in \mathcal{D}er\mathcal{C}txt_1(\mathbf{Ni}+\mathrm{TND})$ holds. ■

4.3.4 Basic Results

It turns out that the basic interconnections stated by Lemma 4.2.4 between the notions of rule correctness, admissibility, and derivability in pure Hilbert systems are preserved in natural-deduction systems for the notions of rule cr-correctness, cr-admissibility, and derivability defined in Definition 4.3.2 (and here rule admissibility is a weaker notion that any of these). This is stated by the following counterpart to Lemma 4.2.4 regarding natural-deduction systems.

Lemma 4.3.5. *Let $\mathcal{S}$ be a natural-deduction system and let R be a natural-deduction system rule on the set of formulas of $\mathcal{S}$. Then the following statements holds:*

(i) R is cr-correct for $\mathcal{S}$ if and only if R is cr-admissible in $\mathcal{S}$.

(ii) If R is derivable in $\mathcal{S}$, then R is also cr-admissible in $\mathcal{S}$. If R is cr-correct for $\mathcal{S}$, then R is also admissible in $\mathcal{S}$. Neither of the two implications in the opposite direction holds in general.

(iii) If R is derivable in $\mathcal{S}$, then R is derivable in every extension by enlargement of $\mathcal{S}$.

An analogous version of this lemma with respect to the concept of ANDS is Lemma B.2.24 in Appendix B.

4.3.5 Rule Elimination

It turns out that cr-admissibility and admissibility of a rule R in a natural-deduction system $\mathcal{S}$ can be characterized, in a similar way as in pure Hilbert systems, in terms of from which derivations in $\mathcal{S}$+R, and in what manner, applications of R can be eliminated (here we understand “can be eliminated” in an analogous sense as explained above just before Theorem 4.2.8). The following theorem can be looked upon as a counterpart to Theorem 4.2.8 in natural-deduction systems; however, it does not cover a statement concerning rule derivability (the reason being that the situation is more complex in this case, cf. further comments below).

Theorem 4.3.6. (Elimination of (cr-)admissible rules). *Let $\mathcal{S}$ be a natural-deduction system with sets Fo and Mk as formulas and assumption markers. And let furthermore R be a rule for $\mathcal{S}$.*

(i) R is admissible in $\mathcal{S}$ if and only if

$$(\forall \mathcal{D} \in \mathcal{D}er(\mathcal{S}{+}R))\left[\mathsf{omassm}(\mathcal{D}) = \emptyset \implies (\exists\, \mathcal{D}' \in \mathcal{D}er(\mathcal{S}))\,[\mathcal{D}' \simeq \mathcal{D}]\right] .$$

holds, i.e. iff the applications of R can be eliminated from every derivation $\mathcal{D}$ in $\mathcal{S}$ that does not contain open assumptions.

(ii) R is cr-admissible in $\mathcal{S}$ if and only

$$(\forall \mathcal{D} \in \mathcal{D}er(\mathcal{S}{+}R))\,(\exists\, \mathcal{D}' \in \mathcal{D}er(\mathcal{S}))\left[\mathcal{D}' \simeq \mathcal{D}\right] .$$

holds, i.e iff applications of R can be eliminated from every derivation $\mathcal{D}$ of $\mathcal{S}$.

The statements (i) and (ii) in Theorem 4.3.6 correspond to the assertions in items (i) and (ii) of Theorem B.2.26 in Appendix B. In Theorem B.2.26, (iii), furthermore a characterization of rule derivability in ANDS's is given which can informally be phrased as follows: a rule R is derivable in a natural-deduction system $\mathcal{S}$ if and only if the applications of R can be eliminated from all derivation contexts in $\mathcal{S}{+}R$ (see Definition B.2.25 in Appendix B for a precise formulation of a notion of rule elimination in ANDS's which corresponds to rule derivability in ANDS's via Theorem B.2.26, (iii)).

Chapter 5

Three Kinds of Proof Systems for Recursive Type Equality

The main aim we have in this chapter is to give precise definitions of three kinds of known proof systems for recursive type equality $=_\mu$. Systems of the first two kinds, to be treated in Section 5.1, are axiom systems that are sound and complete with respect to $=_\mu$. In contrast to this, the systems of the third kind, considered in Section 5.2, are not axiomatizations of $=_\mu$, but are tailor-made for the special purpose of allowing "consistency-checks" to be carried out with respect to $=_\mu$ for arbitrary given equations between recursive types. Apart from defining the main systems of these three kinds, we additionally introduce certain variant systems that will turn out to be very useful for our proof-theoretic investigations in later chapters. In Section 5.3 we are concerned with basic observations about the differences in proof-theoretic properties between the axiom systems in Section 5.1 and the proof systems for consistency-checking in Section 5.2.

5.1 Axiom Systems for Recursive Type Equality

The first formal axiomatization of $=_\mu$ was presented by Amadio and Cardelli in [AmCa93]. Their system is *essentially* the system $\mathbf{AC}^=$ given in the following definition; the only difference between $\mathbf{AC}^=$ and the system given in [AmCa93, Section 5.1, p.30] consists in the fact that the axiom scheme (REN) of $\mathbf{AC}^=$ for taking variants of recursive types is not a formal part of the system given by Amadio and Cardelli. This is because renaming of bound variables in recursive types ("α-conversion" on recursive types) is not dealt with there explicitly; instead, recursive types τ and σ such that $\tau \equiv_{\text{ren}} \sigma$ are identified implicitly on a *syntactical*

Figure 5.1: The Hilbert-style proof system $\mathbf{AC}^=$ for recursive type equality $=_\mu$ given by Amadio and Cardelli.

The *axioms* of $\mathbf{AC}^=$:

$$\text{(REFL)}\ \overline{\tau = \tau} \qquad\qquad \text{(REN)}\ \overline{\tau = \tau'}\ \ (\text{if } \tau \equiv_{\text{ren}} \tau')$$

$$(\mu - \bot)\ \overline{\mu\alpha.\,\alpha = \bot} \qquad\qquad \text{(FOLD/UNFOLD)}\ \overline{\mu\alpha.\,\tau = \tau[\mu\alpha.\,\tau/\alpha]}$$

The *inference rules* of $\mathbf{AC}^=$:

$$\frac{\sigma = \tau}{\tau = \sigma}\ \text{SYMM} \qquad\qquad \frac{\tau = \rho \qquad \rho = \sigma}{\tau = \sigma}\ \text{TRANS}$$

$$\frac{\tau = \sigma}{\mu\alpha.\,\tau = \mu\alpha.\,\sigma}\ \mu\text{-COMPAT} \qquad\qquad \frac{\tau_1 = \sigma_1 \qquad \tau_2 = \sigma_2}{\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2}\ \text{ARROW}$$

$$\frac{\tau_1 = \tau[\tau_1/\alpha] \qquad \tau_2 = \tau[\tau_2/\alpha]}{\tau_1 = \tau_2}\ \text{UFP} \quad (\text{if } \alpha \downarrow \tau)$$

level[1].

Definition 5.1.1 (The proof system $\mathbf{AC}^=$). The proof system $\mathbf{AC}^=$ is defined as a (pure) Hilbert-style proof system in the following way. The *formulas* of $\mathbf{AC}^=$ are the equations between recursive types, i.e. all elements of the set $\mu Tp\text{–}Eq$.

The *axioms* of $\mathbf{AC}^=$ are all those equations between recursive types that belong to one of the four schemes (REFL), (REN), (μ−$\bot$) and (FOLD/UNFOLD) shown in Figure 5.1.

And the *inference rules* of $\mathbf{AC}^=$ are the rules SYMM, TRANS, ARROW, μ-COMPAT and UFP whose respective applications are schematically defined in Figure 5.1. The notation[2] $\alpha \downarrow \tau$ used to denote the necessary side-condition on the applicability of the rule UFP is meant to abbreviate the verbal expression "the recursive type τ is *contractive* in the variable α" that is defined as the condition "$\alpha \notin \text{fv}(\tau)$ or $\tau \equiv \mu\alpha_1 \ldots \alpha_n.\,(\rho_1 \to \rho_2)$ (for $n \in \omega$ and some $\rho_1, \rho_2 \in \mu Tp$ and $\alpha_1 \ldots \alpha_n \in TVar$)" on α and τ. ☒

Remark 5.1.2. In the following three items we gather a few basic observations about the proof system $\mathbf{AC}^=$.

(a) Amadio and Cardelli have introduced the last rule in Figure 5.1 under the name label "(contract)" (see [AmCa93, p.30]). We have, however, chosen

[1] In [AmCa93, p.11] Amadio and Cardelli write "Types are identified up to renaming of bound variables" and, in a footnote, that they use the symbol $\equiv$ except for defining abbreviations also for syntactic identification of recursive types; that is, they use a formal expression $\tau \equiv \sigma$, if this is not intended as introducing τ as an abbreviation for σ, in the meaning of $\tau \equiv_{\text{ren}} \sigma$ according to our notation.

[2] This notation is taken from [AmCa93, p.30].

to call it UFP instead, which is intended to stand short for "unique fixed-point rule", thereby being able to reserve the name "CONTRACT" for a close variant of this rule that is introduced below in (5.1). We think that the formal behaviours of these two rules are better reflected in their names as stipulated here.

A couple of alternative formulations of the "guardedness (side-)condition" $\alpha \downarrow \tau$ in the schematic definition of the applications of the rule UFP in Figure 5.1 will later be given in Chapter 7 (cf. Lemma 7.1.1 and Lemma 7.1.2).

(b) Due to the presence of the axiom scheme (REN) in $\mathbf{AC}^{=}$, the axioms of the scheme (REFL) are actually redundant; but they have been kept here in the intention of taking over the system defined in [AmCa93] as literally as possible for the formulation of a slightly refined system, the system $\mathbf{AC}^{=}$ defined here, in which α-conversion of recursive types is cared for (more[3]) explicitly.

And due to the presence of the transitivity rule TRANS in $\mathbf{AC}^{=}$, the axiom scheme (REN) could have been replaced by one of the rules

$$\frac{\tau' = \sigma}{\tau = \sigma}\,\mathrm{REN}_l \quad (\text{if } \tau' \equiv_{\mathrm{ren}} \tau) \qquad \text{or} \qquad \frac{\tau = \sigma'}{\tau = \sigma}\,\mathrm{REN}_r \quad (\text{if } \sigma' \equiv_{\mathrm{ren}} \sigma)\,.$$

A two-sided version REN of these rules, which allows to take respective variants of recursive types on either side of an equation in its premise and for which its applications can be modeled as juxtapositions of REN_l- and REN_r-applications, will play a part in the proof systems $\mathbf{AK}_{\mathbf{0}}^{=}$ and $\mathbf{HB}_{\mathbf{0}}^{=}$ defined later. The rules REN_l and REN_r will also sometimes be used for the purpose of abbreviating $\mathbf{AC}^{=}$-derivations, since it is obvious, how applications of these rules can be eliminated from an $\mathbf{AC}^{=}$-derivation by replacing them by applications of TRANS-rules with axioms (REN).

(c) Also, in the formulation of $\mathbf{AC}^{=}$ the axiom scheme (FOLD/UNFOLD) could have been replaced by either of the rules

$$\frac{\tau'[\mu\alpha.\,\tau/\alpha] = \sigma}{\mu\alpha.\,\tau = \sigma}\,\mathrm{FOLD}_l \quad (\text{whenever } \tau' \equiv_{\mathrm{ren}} \tau) \qquad \text{or}$$

$$\frac{\tau = \sigma'[\mu\alpha.\,\sigma/\alpha]}{\tau = \mu\alpha.\,\sigma}\,\mathrm{FOLD}_r \quad (\text{whenever } \tau' \equiv_{\mathrm{ren}} \tau)\ ,$$

and it is also easy to see that applications of the rules FOLD_l or FOLD_r can be eliminated from an $\mathbf{AC}^{=}$-derivation using axioms (FOLD/UNFOLD), (REN) and the rules TRANS and SYMM. But it should be noted here, that the rules FOLD_l and FOLD_r allow the "folding" of recursive types in full generality contrary to axioms of the scheme (FOLD/UNFOLD). This is because there exist recursive types τ and type variables α such that $\mu\alpha.\tau$ is *not* substitutible

[3] α-conversion of recursive types must be cared for *only somewhat more* explicitly here because the axioms (REN) allow to do an arbitrary number of renamings of bound variables in a recursive type *in one gathered step*.

for α in τ (for example take $\tau \equiv \beta \to \mu\beta.(\alpha \to \beta)$) and hence recursive types τ and α with the property that $\mu\alpha.\,\tau = \tau[\mu\alpha.\,\tau/\alpha]$ is not an axiom of type (FOLD/UNFOLD) (because then $\tau[\mu\alpha.\,\tau/\alpha]$ is not an admissible substitution expression as explained in Convention 3.3.6).

To allow folding and unfolding in full generality the axiom scheme (FOLD/UNFOLD) could also have been replaced in the formulation of $\mathbf{AC}^{=}$ by the axiom scheme

$$\text{(FOLD/UNFOLD)}' \;\; \mu\alpha.\,\tau = \tau'[\mu\alpha.\,\tau/\alpha] \quad (\text{if } \tau' \equiv_{\text{ren}} \tau)\ .$$

However, it is clear, that due to the presence of axioms (REN) and the rules TRANS in $\mathbf{AC}^{=}$ all axioms of (FOLD/UNFOLD)$'$ are actually derivable in $\mathbf{AC}^{=}$.

Example 5.1.3 (A derivation in $\mathbf{AC}^{=}$). In Example 3.6.2 we have considered the two recursive types

$$\tau_1 \equiv \mu\alpha.\,(\alpha \to \bot) \qquad \text{and} \qquad \sigma_1 \equiv \mu\beta.\,((\beta \to \bot) \to \bot)$$

and we have seen that these two recursive types have the same tree unfolding and hence are strongly equivalent. In the system $\mathbf{AC}^{=}$ it is possible to give the following easy derivation of the equation between recursive types $\tau_1 = \sigma_1$:

$$\text{UFP}\ \dfrac{\dfrac{\dfrac{\text{(FOLD/UNFOLD)}}{\tau_1 = \tau_1 \to \bot} \quad \dfrac{\dfrac{\text{(FOLD/UNFOLD)}}{\tau_1 = \tau_1 \to \bot} \quad \dfrac{\text{(REFL)}}{\bot = \bot}}{\tau_1 \to \bot = (\tau_1 \to \bot) \to \bot}\,\text{ARROW}}{\tau_1 = \underbrace{(\tau_1 \to \bot) \to \bot}_{\equiv ((\gamma\to\bot)\to\bot)\,[\tau_1/\gamma]}}\,\text{TRANS} \qquad \dfrac{\text{(FOLD/UNFOLD)}}{\sigma_1 = \underbrace{(\sigma_1 \to \bot) \to \bot}_{\equiv ((\gamma\to\bot)\to\bot)\,[\sigma_1/\gamma]}}}{\tau_1 = \sigma_1}$$

■

The following theorem, which is also due to Amadio and Cardelli, about the system $\mathbf{AC}^{=}$ states that not only does $\tau =_\mu \sigma$ hold for all equations between recursive types $\tau = \sigma$ that are theorems of $\mathbf{AC}^{=}$ ("$\mathbf{AC}^{=}$ is *sound with respect to* $=_\mu$"), but also the opposite is the case, i.e. that an equation between recursive types $\tau = \sigma$ is derivable in $\mathbf{AC}^{=}$ if τ and σ are strongly equivalent ("$\mathbf{AC}^{=}$ is *complete with respect to* $=_\mu$").

Theorem 5.1.4 (Amadio and Cardelli, '93: Soundness and completeness of the axiom system $\mathbf{AC}^{=}$ with respect to $=_\mu$). *The system* $\mathbf{AC}^{=}$ *is sound and complete with respect to recursive type equality* $=_\mu$*, i.e. for all recursive types* $\tau, \sigma \in \mu Tp$ *the following assertion holds:*

$$\vdash_{\mathbf{AC}^{=}} \tau = \sigma \quad \Longleftrightarrow \quad \tau =_\mu \sigma\ .$$

Reference to a Proof. See Proposition 5.1.1 in [AmCa93, p.30] for the soundness-part and Theorem 5.2.6 in [AmCa93, pp.32,33] for the completeness-part.

There is a widely known alternative for the rule UFP in $\mathbf{AC}^{=}$, namely the rule *we* have decided to call CONTRACT (cf. Remark 5.1.2, (a)) that is defined by the scheme

$$\frac{\tau_1 = \tau[\tau_1/\alpha]}{\tau_1 = \mu\alpha.\,\tau}\,\text{CONTRACT} \quad (\text{if } \alpha \downarrow \tau) \tag{5.1}$$

of its applications (where $\alpha \downarrow \tau$ is defined, for all $\alpha \in \mathit{TVar}$ and $\tau \in \mu\mathit{Tp}$, as in Definition 5.1.1). The *fixed-point rule* CONTRACT is actually just a counterpart to a corresponding rule "Folding", which appears in (the report version of) [ArKl95, p. 39] as part of a complete proof system EL_μ for μ-terms over a first-order signature ([ArKl95] refer in their turn to [Mil84], where a corresponding rule of the same form appears in the context of a "complete inference system for regular behaviours" as R4 on p. 454).

In the context of the axioms and of the rules of $\mathbf{AC}^{=}$ except UFP, the rules UFP and CONTRACT turn out to be equivalent. Since the use of the rule CONTRACT often allows to write down derivations in a more compact form, we will frequently make use of it instead of the rule UFP. Hence there arises the need for a designation of a variant system of $\mathbf{AC}^{=}$ that contains CONTRACT in place of UFP, and for a statement that asserts the existence of easy effective translations between derivations in $\mathbf{AC}^{=}$ and derivations in the variant system that will now be defined.

Definition 5.1.5 (The variant system $\mathbf{AC}^{=}_{*}$ of $\mathbf{AC}^{=}$). The (pure) Hilbert-style proof system $\mathbf{AC}^{=}_{*}$ has the same *formulas* and *axioms* as $\mathbf{AC}^{=}$ and it possesses as *inference rules* precisely the rule CONTRACT with applications of the form (5.1) as well as all rules of $\mathbf{AC}^{=}$ except the rule UFP. ⊠

Proposition 5.1.6 (Equivalence of the systems $\mathbf{AC}^{=}_{*}$ and $\mathbf{AC}^{=}$). *The rule* UFP *is a derivable rule of* $\mathbf{AC}^{=}_{*}$ *and the rule* CONTRACT *is a derivable rule of* $\mathbf{AC}^{=}$*, respective applications of which can be eliminated effectively from derivations in* $\mathbf{AC}^{=}_{*}$+UFP*, or respectively, from derivations in* $\mathbf{AC}^{=}$+CONTRACT*. Hence there are effective transformations from derivations in* $\mathbf{AC}^{=}_{*}$ *into mimicking derivations in* $\mathbf{AC}^{=}$ *and* vice versa.

It follows that the systems $\mathbf{AC}^{=}$ *and* $\mathbf{AC}^{=}_{*}$ *are equivalent, i.e. that they have the same theorems.*

Proof. It is sufficient to show that the rule CONTRACT is a derivable rule of $\mathbf{AC}^{=}$, that the rule UFP is a derivable rule of $\mathbf{AC}^{=}_{*}$ and that furthermore applications of each of these rules can respectively be mimicked with the help of effectively found derivations containing applications of the respective other rule in a very easy way. We will only show that CONTRACT is a derivable rule in $\mathbf{AC}^{=}$, applications of which can be eliminated effectively from arbitrary derivations in $\mathbf{AC}^{=}+$CONTRACT ; the respective statements about UFP are still easier to demonstrate.

For showing that CONTRACT is derivable in $\mathbf{AC}^=$, let ι be an arbitrary application of this rule. We have to show that the derivation $\mathcal{D}_{(\iota)}$ in $\mathbf{AC}^=_*$ that corresponds to this application and that is of the form (5.1), for some $\alpha \in \mathit{TVar}$ and $\tau, \tau_1 \in \mu Tp$ such that $\alpha \downarrow \tau$ holds, can be mimicked by a derivation $\mathcal{D}^{(\iota)}_{\text{mim}}$ in $\mathbf{AC}^=$. Let τ' be a variant of τ such that both $\mu\alpha.\,\tau$ and τ_1 are substitutible for α in τ'. Then also $\mu\alpha.\,\tau \equiv_{\text{ren}} \mu\alpha.\,\tau'$ and $\mu\alpha.\,\tau'$ is substitutible for α in τ'. From a lemma in Chapter 7, Lemma 7.1.2, (i), we conclude that also $\alpha \downarrow \tau'$ holds. Hence the application of UFP in the following derivation $\mathcal{D}^{(\iota)}_{\text{mim}}$ with the assumption $\tau_1 = \tau[\tau_1/\alpha]$

$$\dfrac{\dfrac{\dfrac{\overset{\text{(Assumption)}}{\tau_1 = \tau[\tau_1/\alpha]} \qquad \dfrac{\text{(REN)}}{\tau[\tau_1/\alpha] = \tau'[\tau_1/\alpha]}}{\tau_1 = \tau'[\tau_1/\alpha]} \qquad \dfrac{\text{(FOLD/UNFOLD)}}{\mu\alpha.\,\tau' = \tau'[\mu\alpha.\,\tau'/\alpha]}}{\tau_1 = \mu\alpha.\,\tau'}\,\text{UFP} \qquad \dfrac{\text{(REN)}}{\mu\alpha.\,\tau' = \mu\alpha.\,\tau}}{\tau_1 = \mu\alpha.\,\tau}$$

(where the two applications without name labels of two-premise rules are applications of TRANS) is justified. Now clearly $\mathcal{D}^{(\iota)}_{\text{mim}} \precsim \mathcal{D}_{(\iota)}$ holds, i.e. $\mathcal{D}_{(\iota)}$ is mimicked by the $\mathbf{AC}^=$-derivation $\mathcal{D}^{(\iota)}_{\text{mim}}$.

Let us now consider an arbitrary derivation $\tilde{\mathcal{D}}$ in $\mathbf{AC}^= +\,$CONTRACT, and an arbitrary occurrence of an application ι of CONTRACT corresponding to a derivation $\mathcal{D}_{(\iota)}$ of the form (5.1), for some $\tau, \tau_1 \in \mu Tp$ and $\alpha \in \mathit{TVar}$, within $\tilde{\mathcal{D}}$. Then a derivation $\mathcal{D}^{(\iota)}_{\text{mim}}$ found for the derivation $\mathcal{D}_{(\iota)}$ as above contains in a leaf at its top the premise $\tau_1 = \tau[\tau_1/\alpha]$ of ι and has the same conclusion as ι (and $\mathcal{D}_{(\iota)}$). Therefore $\mathcal{D}^{(\iota)}_{\text{mim}}$ can be used for eliminating the considered occurrence of the application ι of CONTRACT in $\tilde{\mathcal{D}}$ with the result of a derivation with the same conclusion and the same assumptions as $\tilde{\mathcal{D}}$. Similarly, also all other applications of CONTRACT can be eliminated from $\tilde{\mathcal{D}}$ in an effective way such that the result is a derivation in $\mathbf{AC}^=$ with the same conclusion and the same assumptions as $\tilde{\mathcal{D}}$.

□

We continue by giving two examples of derivations in the variant system $\mathbf{AC}^=_*$ of $\mathbf{AC}^=$.

Example 5.1.7 (A derivation in $\mathbf{AC}^=_*$). We consider again the strongly equivalent recursive types τ_1 and σ_1 from Example 3.6.2. A derivation in $\mathbf{AC}^=_*$ with conclusion $\tau_1 = \sigma_1$ is given in Figure 5.2 (the recursive types τ_1 and σ_2 are displayed explicitly at the bottom of the derivation). This derivation is actually (by one occurrence of an axiom) smaller than the $\mathbf{AC}^=$-derivation with the same conclusion that was given in Example 5.1.3. ■

Example 5.1.8 (Another derivation in $\mathbf{AC}^=_*$). As an example of a derivation in $\mathbf{AC}^=_*$, we want to give a derivation for the axioms $(\mu\mu{-}\mu)$ in Lemma 3.8.4, i.e. we want to show

$$\vdash_{\mathbf{AC}^=_*} \mu\alpha.\,\mu\beta.\,\tau = \mu\gamma.\,\tau[\gamma/\alpha, \gamma/\beta]$$

Figure 5.2: Example of a derivation in $\mathbf{AC}^{=}_{*}$.

$$\dfrac{\dfrac{\dfrac{\text{(FOLD/UNFOLD)}}{\tau_1 = \tau_1 \to \bot} \qquad \dfrac{\dfrac{\text{(FOLD/UNFOLD)}}{\tau_1 = \tau_1 \to \bot} \qquad \dfrac{\text{(REFL)}}{\bot = \bot}}{\tau_1 \to \bot = (\tau_1 \to \bot) \to \bot}\,\text{ARROW}}{\tau_1 = \underbrace{(\tau_1 \to \bot) \to \bot}_{\equiv ((\beta\to\bot)\to\bot)\,[\tau_1/\beta]}}\,\text{TRANS}}{\underbrace{\tau_1}_{\equiv \mu\alpha.\,(\alpha\to\bot)} \quad = \quad \underbrace{\sigma_1}_{\equiv \mu\beta.\,((\beta\to\bot)\to\bot)}}\,\text{CONTRACT}$$

for all $\alpha, \beta, \gamma \in \mathit{TVar}$ and $\tau \in \mu\mathit{Tp}$ such that $\alpha \not\equiv \beta$, $\alpha, \beta \in \mathrm{fv}(\tau)$ and $\gamma \notin \mathrm{fv}(\tau) \vee \vee\, \gamma \equiv \alpha \vee \gamma \equiv \beta$.

Let therefore $\alpha, \beta, \gamma \in \mathit{TVar}$ be such that $\alpha, \beta \in \mathrm{fv}(\tau)$, $\alpha \not\equiv \beta$, $\gamma \notin \mathrm{fv}(\tau) \vee \vee\, \gamma \equiv \alpha \vee \gamma \equiv \beta$ and γ substitutible for α and β in τ. From the assumptions about α, β and τ we conclude that τ must be of the form $\mu\alpha_1 \ldots \alpha_n.\,(\rho_1 \to \rho_2)$ for some $n \in \omega$, $\alpha_1, \ldots, \alpha_n \in \mathit{TVar}$ and $\rho_1, \rho_2 \in \mu\mathit{Tp}$ such that $\alpha, \beta \in \mathrm{fv}(\rho_1 \to \rho_2)$ and $\alpha, \beta \not\equiv \alpha_1, \ldots, \alpha_n$; we also let $n \in \omega$ and $\alpha_1, \ldots, \alpha_n, \rho_1, \rho_2$ be chosen in this way.

We furthermore let $\sigma \equiv \mu\gamma.\,\tau[\gamma/\alpha, \gamma/\beta]$ and choose a variant τ' of τ such that σ is substitutible for α and β in τ'; we furthermore choose $\tilde{\gamma} \in \mathit{TVar}$ such that $\tilde{\gamma} \notin \mathrm{fv}(\tau') \vee \tilde{\gamma} \equiv \alpha \vee \tilde{\gamma} \equiv \beta$ such that $\tilde{\gamma}$ is substitutible for α and β in τ' and in τ. Then also $\sigma' \equiv \mu\tilde{\gamma}.\,\tau'[\tilde{\gamma}/\alpha, \tilde{\gamma}/\beta]$ is a variant of σ because of

$$\sigma' \equiv \mu\tilde{\gamma}.\,\tau'[\tilde{\gamma}/\alpha, \tilde{\gamma}/\beta] \equiv_{\mathrm{ren}} \mu\tilde{\gamma}.\,\tau[\tilde{\gamma}/\alpha, \tilde{\gamma}/\beta] \equiv_{\mathrm{ren}}$$
$$\equiv_{\mathrm{ren}} \mu\gamma.\,\tau[\tilde{\gamma}/\alpha, \tilde{\gamma}/\beta][\gamma/\tilde{\gamma}] \equiv_{\mathrm{ren}} \mu\gamma.\,\tau[\gamma/\alpha, \gamma/\beta] \equiv \sigma$$

and furthermore σ' is substitutible for α and β in τ' (because of $\mathrm{fv}(\sigma') = \mathrm{fv}(\sigma)$). Since $\tau' \equiv_{\mathrm{ren}} \tau$, we can conclude (from what we have found about the form of τ above) that $\tau' \equiv_{\mathrm{ren}} \mu\tilde{\alpha}_1 \ldots \tilde{\alpha}_n.\,(\tilde{\rho}_1 \to \tilde{\rho}_2)$ for some $\tilde{\alpha}_1, \ldots, \tilde{\alpha}_n \in \mathit{TVar}$ and $\tilde{\rho}_1, \tilde{\rho}_2 \in \in \mu\mathit{Tp}$ such that $\alpha, \beta \in \mathrm{fv}(\tilde{\rho}_1 \to \tilde{\rho}_2)$ and $\alpha, \beta \not\equiv \tilde{\alpha}_1, \ldots, \tilde{\alpha}_n$.

In this situation, the derivation shown in Figure 5.3 is a derivation in $\mathbf{AC}^{=}_{*}$, where the following justifications for syntactic transformations on substitution expressions and for the applicability of the two occurring applications of CONTRACT have been used:

(I): is due to $\beta \notin \mathrm{fv}(\sigma')$ (since $\sigma' \equiv \mu\tilde{\gamma}.\,\tau'[\tilde{\gamma}/\alpha, \tilde{\gamma}/\beta]$);

(II): $\beta \downarrow \tau'[\sigma'/\alpha]$ holds, since $\tau'[\sigma'/\alpha] \equiv \mu\tilde{\alpha}_1 \ldots \tilde{\alpha}_n.\,(\tilde{\rho}_1[\sigma'/\alpha] \to \tilde{\rho}_2[\sigma'/\alpha])$;

(III): follows from $\alpha \not\equiv \beta$ and $\beta \notin \mathrm{fv}(\sigma')$ with Lemma 3.3.10, (iii);

(IV): $\alpha \downarrow \mu\beta.\,\tau'$ holds, since $\mu\beta.\,\tau' \equiv \mu\beta\tilde{\alpha}_1 \ldots \tilde{\alpha}_n.\,(\tilde{\rho}_1 \to \tilde{\rho}_2)$ and $\alpha \in \mathrm{fv}(\tilde{\rho}_1 \to \tilde{\rho}_2)$ as well as $\alpha \not\equiv \beta, \tilde{\alpha}_1, \ldots \tilde{\alpha}_n$.

Figure 5.3: A derivation in $\mathbf{AC}^{=}_{*}$ of an axiom $(\mu\mu{-}\mu)$.

$$\begin{array}{cl}
\text{(FOLD/UNFOLD)} & \\
\hline
\overbrace{\mu\tilde{\gamma}.\,\tau'}^{\equiv\sigma'}[\tilde{\gamma}/\alpha,\tilde{\gamma}/\beta] = \underbrace{\tau'\,[\sigma'/\alpha,\sigma'/\beta]}_{\underset{(\mathrm{I})}{\Leftrightarrow}\ \tau'[\sigma'/\alpha]\,[\sigma'/\beta]} & \\
\hline
\sigma' = \underbrace{\mu\beta.\,\tau'[\sigma'/\alpha]}_{\underset{(\mathrm{III})}{\Leftrightarrow}\ (\mu\beta.\,\tau')[\sigma'/\alpha]} & \text{CONTRACT}\quad\text{(cf. (II))} \\
\hline
\sigma' = \mu\alpha.\,\mu\beta.\,\tau' & \text{CONTRACT}\quad\text{(cf. (IV))} \\
\hline\hline
\sigma = \mu\alpha.\,\mu\beta.\,\tau & \mathrm{REN}_{l/r} \\
\hline
\mu\alpha.\,\mu\beta.\,\tau = \mu\gamma.\,\tau[\gamma/\alpha,\gamma/\beta] & \text{SYMM}
\end{array}$$

Thus we have recognized that the axioms $(\mu\mu{-}\mu)$ in Lemma 3.8.4 are derivable in $\mathbf{AC}^{=}_{*}$.

A proof for the derivability of the axioms $(\mu\mu{-}\mu)$ in the system $\mathbf{AC}^{=}$ is sketched in [AmCa93, Section 5.1.2, pp.30,31]; there the derivability in $\mathbf{AC}^{=}$ of a substitution rule introduced in Section 7.1, Chapter 7, is used (which rule is not introduced explicitly in [AmCa93], see also Remark 7.1.10). ■

As a direct consequence of Theorem 5.1.4, the soundness and completeness theorem of $\mathbf{AC}^{=}$, and of Proposition 5.1.6 we find the following soundness and completeness theorem for $\mathbf{AC}^{=}_{*}$.

Theorem 5.1.9. *The system* $\mathbf{AC}^{=}_{*}$ *is sound and complete with respect to recursive type equality* $=_{\mu}$.

By using the soundness part of this theorem, we are now eventually able to give the (remaining parts of the) proof of Theorem 3.8.3 from Chapter 3.8.

Proof of Theorem 3.8.3. We know from Lemma 3.8.2 that the transformation ^{c} is a well-defined function from the set μTp of recursive types to the set *can-*μTp of recursive types in canonical form. Hence for proving (3.38) it suffices to establish:

$$(\forall\,\tau \in \mu Tp)\ \ [\,\tau =_{\mu} \tau^{c}\,]\ .$$

For showing this assertion it furthermore suffices, due to the soundness part of the Theorem 5.1.9, to show:

$$(\forall\,\tau \in \mu Tp)\ \ [\,\vdash_{\mathbf{AC}^{=}_{*}}\ \tau = \tau^{c}\,]\ . \tag{5.2}$$

But this statement is now a consequence of Lemma 3.8.4 according to which, for all $\tau \in \mu Tp$, $\vdash_{\mathbf{WEQ}+(\mu\mu-\mu)}\ \tau = \tau^{c}$ holds: this is because the system $\mathbf{AC}^{=}_{*}$ is an extension by enlargement of **WEQ** in which all axioms ($\mu\mu{-}\mu$) are derivable (as we have seen in Example 5.1.8). □

The second kind of proof systems for recursive type equality $=_\mu$ that we will consider are variant systems of a particular axiomatization of the relation $=_\mu$ on the set *can-μTp* of all recursive types in canonical form that was given by M. Brandt and F. Henglein in [BrHe98]. All of these *Brandt-Henglein systems* contain restricted versions of a circular rule FIX. In a formulation for a natural-deduction system, applications of this rule have, at the bottom of derivations $\mathcal{D}$, the form

$$\dfrac{\begin{array}{c}[\tau = \sigma]^u \\ \mathcal{D}_1 \\ \tau = \sigma\end{array}}{\tau = \sigma}\ \text{FIX},\ u \tag{5.3}$$

with $\tau, \sigma \in \mu Tp$ and a subderivation $\mathcal{D}_1$. The rule FIX can be motivated in a coinductive manner and is, at least at first sight, of a very peculiar form: it allows to infer the conclusion $\tau = \sigma$ from itself as (the only) premise, given that $\tau = \sigma$ has already been derived by a derivation $\mathcal{D}_1$ in which $\tau = \sigma$ is also allowed to figure as an assumption an arbitrary number of times. But the conclusion $\tau = \sigma$ in the resulting derivation $\mathcal{D}$ does not depend any more on the use of the same formula in marked assumptions $(\tau = \sigma)^u$ occurring in $\mathcal{D}_1$, since all open marked assumptions of the form $(\tau = \sigma)^u$ have been *discharged* by then at the application of FIX at the bottom of $\mathcal{D}$.

This rule would obviously allow to derive all formulas in *μTp–Eq* as theorems of a system containing FIX if no side-condition were attached to its applications. Hence such an unrestricted FIX-rule is actually useless for building a sound axiomatization of $=_\mu$. But it turns out that suitably restricted forms of the rule FIX (that is, rules with applications (5.3) where an appropriate side-condition is imposed on the subderivation $\mathcal{D}_1$) lead to a sound and complete axiomatization of $=_\mu$ in the context of the axioms and other basic rules from the system of Amadio and Cardelli. For a clear-cut general definition of a rule FIX that only enables sound inferences with respect to $=_\mu$, within a certain variant Brandt-Henglein system defined later in this chapter, we refer to Definition 6.2.1 in Section 6.2 of Chapter 6. And we also refer to Subsection 9.2.1 of Section 9.2 in Chapter 9, where we report on results about a sound version of the rule FIX in the context of the axiom system for recursive type equality given by Brandt and Henglein.

In the axiomatization given by Brandt and Henglein the use of the rule FIX is restricted to the case in which, with respect to the denotations for an application of FIX as in (5.3), the ultimate rule application in $\mathcal{D}_1$ is an application of the composition rule ARROW; in this case no further side-condition on $\mathcal{D}_1$ is needed. However, the rule FIX does not figure itself as an explicit rule in the system given in [BrHe98] and is replaced instead by the rule ARROW/FIX. Hereby an application of ARROW/FIX is an amalgamation of an application of ARROW with an immediately following application of FIX that is not subjected to any *further*[4] side-condition.

The system $\mathbf{HB}^=$ that is defined below is a *natural-deduction formulation* of a very straightforward extension of the axiomatization given by Brandt and Henglein for recursive type equality $=_\mu$ on the set *can-μTp* of *all recursive types in canonical form.* Instead of axiomatizing the relation $=_\mu$ on *can-μTp* (the original system can be found in Figure 4 in [BrHe98, on p.7]), the system $\mathbf{HB}^=$ is able to axiomatize the relation $=_\mu$ on the set *μTp* of *all recursive types.* For this purpose the additional axiom $(\mu\!-\!\bot)'$, an extended version of the axiom $(\mu\!-\!\bot)$ of $\mathbf{AC}^=$, is used which equates recursive types of the form $\mu\alpha_1 \ldots \alpha_n.\,\alpha_1$ with the recursive type $\bot$ that possesses the same tree unfolding. Furthermore, like in the definition of $\mathbf{AC}^=$, we again have taken up axioms (REN) for equating variants of recursive types into the system $\mathbf{HB}^=$. Such axioms do not occur in the system given in [BrHe98], where recursive types that differ only by the names of bound variables are considered to be identified *implicitly* on a syntactical level[5]. As mentioned above, a distinctive role in both the original system of Brandt and Henglein as well as in the system $\mathbf{HB}^=$ defined below is played by the rule ARROW/FIX, at applications of which assumptions of the form of its conclusion can get discharged.

Definition 5.1.10 (The proof system $\mathbf{HB}^=$). The natural-deduction-style proof system $\mathbf{HB}^=$ is defined as follows.

The *formulas* of the formal system $\mathbf{HB}^=$ are precisely the equations between recursive types in *μTp–Eq.* The *axioms* of $\mathbf{HB}^=$ are all those that belong to one of the axiom schemes (REFL), (REN), $(\mu\!-\!\bot)'$ and (FOLD/UNFOLD) gathered in Figure 5.4. Derivations in $\mathbf{HB}^=$ are allowed to start from *marked assumptions* (Assm) of the kind shown in Figure 5.4; such marked assumptions may later be *discharged* at applications of a particular inference rule of $\mathbf{HB}^=$, see below. An occurrence o of a marked assumption $(\tau = \sigma)^u$ at the top of derivation $\mathcal{D}$ in $\mathbf{HB}^=$ is called an *open* (or *undischarged*) (marked) assumption in $\mathcal{D}$ if and only if on the thread from the occurrence o downwards to the conclusion of $\mathcal{D}$ no rule application is passed at which the occurrence o of $(\tau = \sigma)^u$ is discharged.

As its *inference rules* the system $\mathbf{HB}^=$ possesses precisely the rules SYMM, TRANS and ARROW/FIX whose respective applications are schematically exhibited in Figure 5.4. In contrast to applications of SYMM and TRANS, applications of the rule ARROW/FIX allow to discharge open assumptions: in the derivation $\mathcal{D}$ displayed in Figure 5.4 with an application of ARROW/FIX at its bottom and with immediate subderivations $\mathcal{D}_1$ and $\mathcal{D}_2$, the equations in angle brackets $[\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2]^u$ denote the class of all open assumptions of the form $(\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2)^u$ in $\mathcal{D}_1$ or in $\mathcal{D}_2$, respectively; these two open assumption classes in $\mathcal{D}_1$ and $\mathcal{D}_2$ correspond together and in an obvious way (suggested by the schematic prooftree for $\mathcal{D}$) to a class of assumptions of the form $(\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2)^u$ in $\mathcal{D}$ that commonly *are discharged* at the application of ARROW/FIX at the bot-

[4] Except the stated one that such an application of FIX has to be immediately preceded by an application of ARROW.

[5] "…, recursive types that differ only in their bound variables are identified, …" [BrHe98, p. 3].

Figure 5.4: The natural-deduction-style proof system $\mathbf{HB}^{=}$ following the axiomatization of recursive type equality $=_\mu$ by Brandt and Henglein.

The *axioms* and possible *marked assumptions* in $\mathbf{HB}^{=}$:

$$(\text{REFL})\ \overline{\tau=\tau} \qquad\qquad (\text{REN})\ \overline{\tau=\tau'}\quad (\text{if } \tau\equiv_{\text{ren}}\tau')$$

$$(\mu-\bot)'\ \overline{\mu\alpha\alpha_1\ldots\alpha_n.\,\alpha=\bot} \qquad (\text{FOLD/UNFOLD})\ \overline{\mu\alpha.\,\tau=\tau[\mu\alpha.\,\tau/\alpha]}$$

$$(\text{Assm})\ \ (\tau=\sigma)^u\quad (\text{with } u\in \mathit{Mk})$$

The *inference rules* of $\mathbf{HB}^{=}$:

$$\cfrac{\begin{array}{c}\mathcal{D}_1\\ \sigma=\tau\end{array}}{\tau=\sigma}\ \text{SYMM} \qquad\qquad \cfrac{\begin{array}{cc}\mathcal{D}_1 & \mathcal{D}_2\\ \tau=\rho & \rho=\sigma\end{array}}{\tau=\sigma}\ \text{TRANS}$$

$$\cfrac{\begin{array}{cc}[\tau_1\to\tau_2=\sigma_1\to\sigma_2]^u & [\tau_1\to\tau_2=\sigma_1\to\sigma_2]^u\\ \mathcal{D}_1 & \mathcal{D}_2\\ \tau_1=\sigma_1 & \tau_2=\sigma_2\end{array}}{\tau_1\to\tau_2=\sigma_1\to\sigma_2}\ \text{ARROW/FIX},\ u$$

tom of $\mathcal{D}$; thereby the marker u of the discharged assumptions $[\tau_1\to\tau_2=\sigma_1\to\sigma_2]^u$ is also attached to the particular application of ARROW/FIX at which these assumptions are discharged. Here and in general the markers act as bookkeeping devices that allow to identify which open assumptions get actually discharged at a considered application of ARROW/FIX in a $\mathbf{HB}^{=}$-derivation.

A formula $\tau=\sigma$ is a *theorem* of $\mathbf{HB}^{=}$ (in symbolic notation: $\vdash_{\mathbf{HB}^{=}}\ \tau=\sigma$) iff there exists a derivation $\mathcal{D}$ in $\mathbf{HB}^{=}$ with conclusion $\tau=\sigma$ and with the property that all marked assumptions have been discharged at respective applications of ARROW/FIX in $\mathcal{D}$.

⊠

Remark 5.1.11. (i) If the set of formulas of the system $\mathbf{HB}^{=}$ is restricted to the set *can-μTp–Eq* of equations between recursive types in canonical form, then the axioms $(\mu-\bot)'$ of $\mathbf{HB}^{=}$ do not occur any more in the restricted system and in this manner essentially a natural-deduction formulation[6] of the system given by Brandt and Henglein is obtained.[7]

(ii) As Brandt and Henglein point out in [BrHe98], it is a particular feature of their system that known efficient algorithms for deciding strong recursive type

[6] [BrHe98] present their system in a Gentzen-style sequent-calculus.

[7] By "essentially [...] is obtained" we mean that a system is reached which differs from the one given in [BrHe98] only by the additional presence of the axioms (REN).

equality $=_\mu$ can be modified in such a way as to be able to construct derivations in their system efficiently. But the main advantage of the formal systems introduced by Brandt and Henglein only becomes fully apparent in the case of their related axiomatization for the subtyping relation $\leq_\mu$ on recursive types, which is also given in [BrHe98]. This is because that axiomatization does lend itself to become rebuilt in a straightforward way into an annotated system, in which the annotations c in formulas $c : \tau \leq \sigma$ denote abstract *coercions* (i.e. type-adaptation functions) between recursive types τ and σ. In the case of a formula $c : \tau \leq \sigma$ that is derivable in such an appropriate annotated version of the Brandt-Henglein system for $\leq_\mu$ without open assumptions, the coercion c does also witness that τ and σ are in fact in the subtype relation or—in other words—that τ is a subtype of σ. (This contrasts with the fact that from the outset it is not clear at all how a calculus for subtyping inequations annotated with coercions could be built from an axiomatization of $\leq_\mu$ given by Amadio and Cardelli in [AmCa93]).

For mainly typographical reasons, we do not give an example of a derivation in the system $\mathbf{HB}^{=}$ here. In Example 5.1.14 below, however, we will present a derivation in a close variant system $\mathbf{HB}_{\mathbf{0}}^{=}$ of $\mathbf{HB}^{=}$ ($\mathbf{HB}_{\mathbf{0}}^{=}$ will be introduced in Definition 5.1.13) that can be transformed in a straightforward way into a derivation in the system $\mathbf{HB}^{=}$.[8]

The next theorem, due to Brandt and Henglein, states that $\mathbf{HB}^{=}$ is a complete proof system for recursive type equality $=_\mu$.

Theorem 5.1.12 (Brandt, Henglein, '98: Soundness and completeness of the axiom system $\mathbf{HB}^{=}$ with respect to $=_\mu$). *The system* $\mathbf{HB}^{=}$ *is sound and complete with respect to recursive type equality* $=_\mu$ *; this means that for all recursive types* $\tau, \sigma \in \mu Tp$ *the following statement holds:*

$$\vdash_{\mathbf{HB}^{=}} \tau = \sigma \quad\Longleftrightarrow\quad \tau =_\mu \sigma\,. \tag{5.4}$$

Sketch of Proof. A very direct and straightforward way of *proving the soundness* of the system $\mathbf{HB}^{=}$ with respect to $=_\mu$ (i.e. for showing "$\Rightarrow$" in (5.4)) consists in showing the statement

$$\left.\begin{array}{l} \big(\forall p \in \{1,2\}^*\big)\,\big(\forall \tau,\sigma,\tau_0,\sigma_0 \in \mu Tp\big) \\ \left(\forall \mathcal{D}\ \ \begin{array}{l}\mathbf{HB}^{=}\text{-derivation without open}\\ \text{assumption-classes and of the form}\end{array} \quad \begin{array}{c}\mathcal{D}_0\\ (\tau_0=\sigma_0)\\ \mathcal{DC}\\ \tau=\sigma\end{array}\right) \\ \qquad\qquad\qquad\qquad \Big[\mathbf{Tree}(\tau_0)(p) = \mathbf{Tree}(\sigma_0)(p)\Big]\,. \end{array}\right\} \tag{5.5}$$

[8] For the easy examples treated here, the variant system $\mathbf{HB}_{\mathbf{0}}^{=}$ of $\mathbf{HB}^{=}$, which will be defined below, enables "more compact" derivations of smaller *size* than the system $\mathbf{HB}^{=}$. However, this is not true in general for larger examples, for which due to substantial reasons the reversed situation is possible to occur.

by induction on the length $|p|$ of the path $p \in \{1,2\}^*$ with a sub-induction on the depth $|\mathcal{D}_0|$ of the subderivation $\mathcal{D}_0$ of $\mathcal{D}$. The statement (5.5) obviously implies that $\mathbf{Tree}(\tau) = \mathbf{Tree}(\sigma)$ and hence that $\tau =_\mu \sigma$ holds for every derivation $\mathcal{D}$ in $\mathbf{HB}^=$ without open assumption classes and with conclusion $\tau = \sigma$ for some $\tau, \sigma \in \mu Tp$.

For the *proof of the completeness* of $\mathbf{HB}^=$ with respect to $=_\mu$ (i.e. for a proof of "$\Leftarrow$" in (5.4)) we refer to [BrHe98]. Although there an axiom system for recursive types in canonical form is considered, the method of proof used by Brandt and Henglein can easily be adapted for our (slightly extended[9]) system $\mathbf{HB}^=$ to cover the case of recursive type equality $=_\mu$ for general recursive types in μTp (the additional axioms $(\mu-\bot)'$ in $\mathbf{HB}^=$ are essential for this). – We shall not repeat this proof here, since we will have to use its basic idea (and we will in a way repeat the entire proof) in similar circumstances in Chapter 7 in Lemma 7.2.9 (there also the necessary adaptation of the proof in [BrHe98] for the case of our system $\mathbf{HB}^=$ with respect to general recursive types plays a role and will be treated explicitly). □

For our proof-theoretic investigations in coming chapters it will be important to have also a variant system $\mathbf{HB}_\mathbf{0}^=$ of $\mathbf{HB}^=$ with stronger proof-theoretical properties at our disposal that does not contain symmetry and transitivity rules. Below we define such a variant system $\mathbf{HB}_\mathbf{0}^=$ of $\mathbf{HB}^=$ and subsequently give an example of a derivation in this system. In the absence of the rules SYMM and TRANS from this variant system $\mathbf{HB}_\mathbf{0}^=$ of $\mathbf{HB}^=$, the axioms (REN), $(\mu-\bot)'$ and (FOLD/UNFOLD) of $\mathbf{HB}^=$ have to be reformulated as respective rules REN, $(\mu-\bot)_l^{\bot\mathrm{der}}$, $(\mu-\bot)_r^{\bot\mathrm{der}}$, FOLD$_l$ and FOLD$_r$ in $\mathbf{HB}_\mathbf{0}^=$.

A further, but minor, difference between $\mathbf{HB}^=$ and its variant system $\mathbf{HB}_\mathbf{0}^=$ to be defined below consists in the fact that in $\mathbf{HB}_\mathbf{0}^=$ applications of the rule ARROW/FIX are subjected to the side-condition that the class of assumptions that get discharged at applications of this rule is actually non-empty or, as we shall sometimes say, *inhabited*. This has the consequence that in $\mathbf{HB}_\mathbf{0}^=$ additionally the rule ARROW of $\mathbf{AC}^=$ is needed to mimic those applications of ARROW/FIX in $\mathbf{HB}^=$ at which *no* open assumptions are discharged. Our aim with introducing in $\mathbf{HB}_\mathbf{0}^=$ a clear distinction between applications of ARROW and applications of ARROW/FIX consists in making our notation match with certain case-distinctions in proof-theoretic arguments used later.

Definition 5.1.13. (The variant system $\mathbf{HB}_\mathbf{0}^=$ of $\mathbf{HB}^=$). The proof system $\mathbf{HB}_\mathbf{0}^=$ is a natural-deduction-style system; it contains as its *formulas* precisely the equations between recursive types in *μTp–Eq*, it allows *axioms* (REFL), *marked assumptions* (Assm) and possesses as *inference rules* precisely the rules REN, $(\mu-\bot)_l^{\bot\mathrm{der}}$, $(\mu-\bot)_r^{\bot\mathrm{der}}$, FOLD$_l$, FOLD$_r$, ARROW and ARROW/FIX exhibited in Figure 5.5. The notions of *open* or *discharged* occurrences of marked assumptions in a derivation in $\mathbf{HB}_\mathbf{0}^=$ are defined analogously as for the system $\mathbf{HB}^=$ in Definition 5.1.10. The precise meaning of the side-condition $\mathbf{I}$ for applications of the

[9] Our *slight extension* $\mathbf{HB}^=$ of the system considered by Brandt and Henglein was chosen in such a way as to be able to constitute an axiomatization for $=_\mu$ with respect to recursive types in μTp ([BrHe98] do only consider an axiom system with respect to canonical recursive types, i.e. elements of *can-μTp*).

Figure 5.5: The natural-deduction-style proof system $\mathbf{HB}_\mathbf{0}^=$ for recursive type equality $=_\mu$, a 'normalized' version without symmetry and transitivity rules of the Brandt-Henglein system $\mathbf{HB}^=$.

The *axioms* and possible *marked assumptions* in $\mathbf{HB}_\mathbf{0}^=$:

$$(\text{REFL})\;\; \dfrac{}{\tau = \tau} \qquad\qquad (\text{Assm})\;\; (\tau = \sigma)^u \quad (\text{with } u \in \mathit{Mk})$$

The *inference rules* of $\mathbf{HB}_\mathbf{0}^=$:

$$\begin{array}{c}\mathcal{D}_1\\ \dfrac{\perp = \sigma}{\mu\alpha\alpha_1\dots\alpha_n.\,\alpha = \sigma}\end{array}\;(\mu{-}\perp)^{\perp\text{der}}_l \qquad\qquad \begin{array}{c}\mathcal{D}_1\\ \dfrac{\tau = \perp}{\tau = \mu\beta\beta_1\dots\beta_n.\,\beta}\end{array}\;(\mu{-}\perp)^{\perp\text{der}}_r$$

$$\begin{array}{c}\mathcal{D}_1\\ \dfrac{\tau[\mu\alpha.\,\tau/\alpha] = \sigma}{\mu\alpha.\,\tau = \sigma}\end{array}\;\text{FOLD}_l \qquad\qquad \begin{array}{c}\mathcal{D}_1\\ \dfrac{\tau = \sigma[\mu\beta.\,\sigma/\beta]}{\tau = \mu\beta.\,\sigma}\end{array}\;\text{FOLD}_r$$

$$\begin{array}{c}\mathcal{D}_1\\ \dfrac{\tau = \sigma}{\tau' = \sigma'}\end{array}\;\text{REN}\quad \begin{array}{l}(\text{if } \tau' \equiv_{\text{ren}} \tau\\ \;\text{and } \sigma' \equiv_{\text{ren}} \sigma)\end{array} \qquad\qquad \dfrac{\begin{array}{cc}\mathcal{D}_1 & \mathcal{D}_2\\ \tau_1 = \sigma_1 & \tau_2 = \sigma_2\end{array}}{\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2}\;\text{ARROW}$$

$$\dfrac{\begin{array}{cc}[\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2]^u & [\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2]^u\\ \mathcal{D}_1 & \mathcal{D}_2\\ \tau_1 = \sigma_1 & \tau_2 = \sigma_2\end{array}}{\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2}\;\begin{array}{l}\text{ARROW/FIX}, u\\ \quad(\text{if side-cond. } \mathbf{I})\end{array}$$

rule ARROW/FIX in $\mathbf{HB}_\mathbf{0}^=$ consists—with respect to the denotations used in the displayed form of this rule in Figure 5.5—in the statement:

> "The class of open assumptions of the form $(\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2)^u$ is inhabitated (non-empty) in *at least one* of the derivations $\mathcal{D}_1$ or $\mathcal{D}_2$."

For informal references in arguments, proofs, etc., we will allow 'bundling' respective left- and right-sided rules: by speaking about rules $\text{FOLD}_{l/r}$, we will refer to both of the rules FOLD_l and FOLD_r; and by rules $(\mu{-}\perp)^{\perp\text{der}}_{l/r}$ we will mean the rules $(\mu{-}\perp)^{\perp\text{der}}_l$ and $(\mu{-}\perp)^{\perp\text{der}}_r$.

Furthermore a formula $\tau = \sigma$ is a *theorem* of $\mathbf{HB}_\mathbf{0}^=$ (which is symbolically denoted by $\vdash_{\mathbf{HB}_\mathbf{0}^=} \tau = \sigma$) if and only if there exists a derivation $\mathcal{D}$ in $\mathbf{HB}_\mathbf{0}^=$ with conclusion $\tau = \sigma$ and with the property that all marked assumptions have been discharged at respective applications of ARROW/FIX in $\mathcal{D}$.

⊠

We continue with an example in which we develop a derivation in $\mathbf{HB}_\mathbf{0}^=$.

Example 5.1.14 (A derivation in $\mathbf{HB}_{\mathbf{0}}^{=}$). We look again at the two strongly equivalent recursive types τ_1 and σ_1 from Example 3.6.2 (which can be found explicitly here at the bottom of the derivation below). If one tries to build a derivation in the system $\mathbf{HB}_{\mathbf{0}}^{=}$ from the desired conclusion $\tau_1 = \sigma_1$ upwards, and if one thereby avoids unnecessary applications of REN and discharges arising assumptions always as soon as possible, then one arrives at the following shortest possible derivation $\mathcal{D}$ of $\tau_1 = \sigma_1$ in $\mathbf{HB}_{\mathbf{0}}^{=}$:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\left(\tau_1 \to \bot = (\sigma_1 \to \bot) \to \bot\right)^u}{\tau_1 = \sigma_1}\;\text{FOLD}_{l/r} \qquad \dfrac{(\text{REFL})}{\bot = \bot}}{\tau_1 \to \bot = \sigma_1 \to \bot}\;\text{ARROW}}{\tau_1 = \sigma_1 \to \bot}\;\text{FOLD}_l \qquad \dfrac{(\text{REFL})}{\bot = \bot}}{\tau_1 \to \bot = (\sigma_1 \to \bot) \to \bot}\;\text{ARROW/FIX},\,u}{\underbrace{\mu\alpha.\,(\alpha \to \bot)}_{\equiv\,\tau_1} = \underbrace{\mu\beta.\,((\beta \to \bot) \to \bot)}_{\equiv\,\sigma_1}}\;\text{FOLD}_{l/r}$$

The derivation $\mathcal{D}$ contains only five applications of rules that are not *also* part of the system $\mathbf{HB}^{=}$. These are the two pairs of applications of rules $\text{FOLD}_{l/r}$ that are located in $\mathcal{D}$ respectively below the marked assumption $\left(\tau_1 \to \bot = (\sigma_1 \to \bot) \to \bot\right)^u$ at the top and at the bottom, and the application of FOLD_r just above the left premise of the single application of ARROW/FIX in $\mathcal{D}$. By eliminating these five rule applications in the easy way that is described below in the proof of Lemma 5.1.19, the derivation $\mathcal{D}$ can easily be transformed into a corresponding derivation $\mathcal{D}'$ in the system $\mathbf{HB}^{=}$ (which instead of the five applications of rules $\text{FOLD}_{l/r}$ contains additionally five occurrences of axioms (FOLD/UNFOLD) as well as five applications of TRANS and two applications of SYMM). ■

Apart from its usefulness in proof-theoretic transformations developed in later chapters, the principal motivation for considering the variant system $\mathbf{HB}_{\mathbf{0}}^{=}$ of $\mathbf{HB}^{=}$ consists, as hinted earlier, in the desire to have a system available that is equivalent to $\mathbf{HB}^{=}$, but that has stronger proof-theoretical properties. This is because a considerable degree of complexity in the search for derivations in $\mathbf{HB}^{=}$ is due to the presence of the rule TRANS and, to a much lesser extent, the presence of the rule SYMM in $\mathbf{HB}^{=}$. These two rules, however, are not part of the variant system $\mathbf{HB}_{\mathbf{0}}^{=}$.

In fact, the system $\mathbf{HB}_{\mathbf{0}}^{=}$ can be considered as a 'normalized' version[10] of the system $\mathbf{HB}^{=}$. Some obvious effects of applications of the symmetry and transitivity rules in $\mathbf{HB}^{=}$ are formalized through the rules REN, $(\mu{-}\bot)_{l/r}^{\bot\text{der}}$ and $\text{FOLD}_{l/r}$ in

[10] Brandt and Henglein use the expression 'normalized inference rules' for related systems concerned with the subtyping relation on recursive types in Figures 7 and 8 on pp. 20 and 21 in [BrHe98], which they use for explaining the coinductive background for their axiomatization of $=_\mu$.

$\mathbf{HB}_{\mathbf{0}}^{=}$. The resulting SYMM- and TRANS-free system is still sound and complete with respect to $=_\mu$, as we will see shortly. And furthermore, we will demonstrate later in Chapter 8, Section 8.2, that every derivation in $\mathbf{HB}^{=}$ without open assumption classes can effectively be transformed into a derivation of $\mathbf{HB}_{\mathbf{0}}^{=}$ with the same conclusion and without open assumptions. The process underlying this transformation (cf. Section 8.2 in Chapter 8) proceeds by 'working away' if not eliminating applications of SYMM and TRANS and can be thought of as a process of *normalization* for $\mathbf{HB}^{=}$-derivations.

Still further justification for calling the system $\mathbf{HB}_{\mathbf{0}}^{=}$ a 'normalized' version of $\mathbf{HB}^{=}$ consists in the fact that derivations in $\mathbf{HB}_{\mathbf{0}}^{=}$ satisfy a version of the *subformula property*, which is a very desirable feature for proof-theoretical investigations of a proof system. Such a property is obeyed by many sequent-style calculi, "Gentzen systems" in the formulation of [TS00], that do not contain the cut rule, but that admit cut-elimination. Usually, it is said of a Gentzen-system $\mathcal{G}$ that it *fulfills the subformula property* if and only if in every derivation of a sequent $\Gamma \Rightarrow \Delta$ only subformulas of Γ and Δ occur (depending on the Gentzen system $\mathcal{G}$, Γ and Δ are finite sets, finite multisets or finite sequences of formulas). A related, but formally different, kind of subformula property is encountered for derivations in some natural-deduction systems, like for example the natural-deduction system **Ni** for intuitionistic predicate logic formulated in [TS00]. There it is fulfilled only by 'normal' derivations (cf. Definition 6.1.2 in [TS00, p.179]) and has the following form: every formula that occurs in a normal derivation $\mathcal{D}$ of **Ni** is either a subformula of the conclusion of $\mathcal{D}$ or a subformula of one of the open assumptions in $\mathcal{D}$ (Theorem 6.2.7 in [TS00, p.188]).

For the formulation of a subformula property that is appropriate for systems with equations between recursive types as formulas, we need a suitable notion of 'subformula' of a formula in $\mathbf{HB}_{\mathbf{0}}^{=}$ that abstracts away slightly from the usual notion of subformula of a formula in predicate logic (as defined, for instance, by the notion of a "Gentzen subformula" in [TS00, Section 1.1.3, p.4]). The reason being that the formulas of $\mathbf{HB}_{\mathbf{0}}^{=}$, the equations from the set $\mu Tp\text{–}Eq$, are *atomic* formulas of predicate logic with equality, which do not have proper subformulas.

Definition 5.1.15 ('Subformulas' of equations between recursive types). Let τ, σ, χ_1 and χ_2 be recursive types. The equation $\chi_1 = \chi_2$ is a *'subformula'* of the equation $\tau = \sigma$ if and only if $\chi_1 \sqsubseteq' \tau$ and $\chi_2 \sqsubseteq' \sigma$ hold, i.e. iff χ_1 and χ_2 are $\rightarrow_{\text{roud}\perp}$-generated subterms (cf. Definition 3.9.20) of, respectively, τ_1 and τ_2. ☒

With this definition we can now define the subformula property SP_1, which is satisfied for $\mathbf{HB}_{\mathbf{0}}^{=}$, as stated by the subsequent proposition.

Definition 5.1.16 (The subformula property SP_1). Let $\mathcal{S}$ be a Hilbert-style or natural-deduction-style proof system with $\mu Tp\text{–}Eq$ as its set of formulas.

We say that a derivation $\mathcal{D}$ in $\mathcal{S}$ with possibly open assumptions *satisfies*, or *fulfills*, *the subformula property* SP_1 if and only if each formula that occurs in $\mathcal{D}$ is a 'subformula' of the conclusion of $\mathcal{D}$. And we say that $\mathcal{S}$ *has*, or *obeys*, *the subformula property* SP_1, or that SP_1 *holds in* $\mathcal{S}$ if and only if every derivation in $\mathcal{S}$ satisfies SP_1. ☒

Proposition 5.1.17. *The system* $\mathbf{HB_0^=}$ *obeys the subformula property* SP_1.

Proof. This follows by a straightforward proof by induction on the depth of derivations in $\mathbf{HB_0^=}$. For the induction step it is used, on the one hand, that the $\to_{\text{roud}\bot}$-generated-subterm relation $\sqsubseteq'$ is transitive, and on the other hand, that for every application $\dfrac{\tilde{\tau}_1 = \tilde{\sigma}_1 \qquad (\ \tilde{\tau}_2 = \tilde{\sigma}_2\ )}{\tilde{\tau} = \tilde{\sigma}}\,R$ of an arbitrary one- (or two-) premise rule R in $\mathbf{HB_0^=}$ $\tilde{\tau}_1 \sqsubseteq' \tilde{\tau}$ and $\tilde{\sigma}_1 \sqsubseteq' \tilde{\sigma}$ (as well as $\tilde{\tau}_2 \sqsubseteq' \tilde{\tau}$ and $\tilde{\sigma}_2 \sqsubseteq' \tilde{\sigma}$) holds. □

Remark 5.1.18 (The subformula property SP_1 **does neither hold in** $\mathbf{HB^=}$, $\mathbf{AC^=}$, $\mathbf{AC_*^=}$, **nor in WEQ).** Due to the presence of the rule TRANS in $\mathbf{HB^=}$, these systems *do not obey* the subformula property SP_1. To see this, we consider the example of an application of TRANS forming the derivation $\mathcal{D}$ in $\mathbf{HB^=}$ of the form

$$\frac{(\tau_2 = \rho_2)^u \qquad (\rho_2 = \sigma_2)^v}{\tau_2 = \sigma_2}\,\text{TRANS} \tag{5.6}$$

with assumption markers u and v and with the strongly equivalent recursive types

$$\tau_2 \equiv \mu\alpha.\,((\alpha \to \alpha) \to \alpha)\ , \quad \rho_2 \equiv \mu\alpha.\,(\alpha \to \alpha)\ , \quad \sigma_2 \equiv \mu\alpha.\,(\alpha \to (\alpha \to \alpha)) \tag{5.7}$$

from Example 3.6.3. It is easy to check that $\rho_2 \not\sqsubseteq' \sigma_2$ and $\rho_2 \not\sqsubseteq' \tau_2$ hold. Hence neither of the assumptions $\tau_2 = \rho_2$ and $\rho_2 = \sigma_2$ of $\mathcal{D}$ is a 'subformula' of the conclusion $\tau_2 = \sigma_2$ of the $\mathbf{HB^=}$-derivation $\mathcal{D}$. Consequently, $\mathbf{HB^=}$ does not obey the subformula property SP_1.

We notice furthermore that the subformula property SP_1 is not just violated only by $\mathbf{HB^=}$-derivations *with open assumptions* as the one in (5.6). Actually, there exist derivations in $\mathbf{HB^=}$ *without open assumptions* and with (5.6), for τ_2, ρ_2, and σ_2 as in (5.7), as their bottommost rule application: in view of Theorem 5.1.12, the completeness theorem of $\mathbf{HB^=}$ with respect to $=_\mu$, this is a consequence of the fact that τ_2, ρ_2, and σ_2 as in (5.7) are strongly equivalent.

It can be shown in an analogous way that the subformula property SP_1 does not hold for proof systems that contain the transitivity rule and that possess μTp–Eq as their set of formulas: for example SP_1 does not hold in **WEQ**, $\mathbf{AC^=}$, and $\mathbf{AC_*^=}$.

The following lemma formalizes an easy observation about the proof-theoretic relationship between the systems $\mathbf{HB_0^=}$ and $\mathbf{HB^=}$, namely, that every derivation in the variant system $\mathbf{HB_0^=}$ can effectively be transformed into a derivation in $\mathbf{HB^=}$ with the same conclusion.

Lemma 5.1.19. *All rules of the system* $\mathbf{HB_0^=}$ *are derivable rules of the system* $\mathbf{HB^=}$. *Furthermore, every derivation* $\mathcal{D}$ *in* $\mathbf{HB_0^=}$ *with possibly open assumption classes can be transformed effectively into a derivation* $\mathcal{D}'$ *in* $\mathbf{HB^=}$ *with the same conclusion and the same (if any) open assumption classes. In particular, for all* $\tau, \sigma \in \mu Tp$

$$\vdash_{\mathbf{HB_0^=}} \tau = \sigma \quad \Longrightarrow \quad \vdash_{\mathbf{HB^=}} \tau = \sigma \tag{5.8}$$

holds, i.e. $\mathbf{HB^=}$ *is an extension of* $\mathbf{HB_0^=}$.

Proof. We are going to prove separately in the two items below that all rules of $\mathbf{HB}^{=}_{\mathbf{0}}$ are derivable in $\mathbf{HB}^{=}$, and that there exists an easy effective transformation of $\mathbf{HB}^{=}_{\mathbf{0}}$-derivations into $\mathbf{HB}^{=}$-derivations.

(a) It is obvious that the rule ARROW is a derivable rule of $\mathbf{HB}^{=}$ because all of its applications correspond to such applications of the rule ARROW/FIX in $\mathbf{HB}^{=}$ at which no assumptions are discharged. The rule ARROW/FIX of $\mathbf{HB}^{=}_{\mathbf{0}}$ is a restricted form of the rule ARROW/FIX of $\mathbf{HB}^{=}$ and hence is derivable in $\mathbf{HB}^{=}$, too.

That the rules REN, $(\mu-\bot)^{\bot\text{der}}_{l/r}$, and $\text{FOLD}_{l/r}$ of $\mathbf{HB}^{=}_{\mathbf{0}}$ are derivable rules of $\mathbf{HB}^{=}$ can respectively be demonstrated in very similar ways. For instance in the case of a rule $(\mu-\bot)^{\text{der}\bot}_{r}$, an arbitrary application ι of this rule at the bottom of a derivation $\mathcal{D}$ of the form

$$\dfrac{\begin{array}{c}\mathcal{D}_1\\ \tau=\bot\end{array}}{\tau=\mu\beta\beta_1\dots\beta_n.\,\beta}\;(\mu-\bot)^{\bot\text{der}}_{r}$$

in $\mathcal{D}er(\mathbf{HB}^{=}+(\mu-\bot)^{\bot\text{der}}_{r})$, with some recursive type τ, and type variables $\beta_1,\dots,\beta_n$ as well as with a derivation $\mathcal{D}_1\in\mathcal{D}er(\mathbf{HB}^{=})$, can be replaced, using a mimicking derivation context $\mathcal{DC}^{(\iota)}_{\text{mim}}\in\mathcal{D}er\mathcal{C}txt_1(\mathbf{HB}^{=})$ that involves the axiom $(\mu-\bot)'$ and the rules SYMM and TRANS of $\mathbf{HB}^{=}$, by a derivation $\tilde{\mathcal{D}}$ with the same open assumptions as $\mathcal{D}$ that is of the form

$$\begin{array}{c}\mathcal{D}_1\\ [\tau=\bot]_1\\ \mathcal{DC}^{(\iota)}_{\text{mim}}\\ \tau=\mu\beta\beta_1\dots\beta_n.\,\beta\end{array}\qquad\text{where } \mathcal{DC}^{(\iota)}_{\text{mim}} \text{ is:}\qquad \dfrac{[\,]_1\qquad\dfrac{\dfrac{\scriptstyle(\mu-\bot)'}{\mu\beta\beta_1\dots\beta_n.\,\beta=\bot}}{\bot=\mu\beta\beta_1\dots\beta_n.\,\beta}\,\text{SYMM}}{\tau=\mu\beta\beta_1\dots\beta_n.\,\beta}\,\text{TRANS}$$

Similar transformations of a mostly even easier kind can be effected for rules REN, $(\mu-\bot)^{\bot\text{der}}_{l}$ and $\text{FOLD}_{l/r}$ to obtain derivations in $\mathbf{HB}^{=}$ that mimic applications of these other rules in $\mathbf{HB}^{=}$.

(b) Let $\mathcal{D}$ be an arbitrary derivation in $\mathbf{HB}^{=}_{\mathbf{0}}$ with conclusion $\tau=\sigma$ for some $\tau,\sigma\in\mu Tp$ and possibly with open assumption classes. The derivation $\mathcal{D}$ can be transformed into a derivation $\mathcal{D}'$ in $\mathbf{HB}^{=}$ with the same conclusion and the same open assumption classes by performing to $\mathcal{D}$, and then to the intermediary results, all respective single elimination steps of the two items below.

   (i) Replace each application of the rule ARROW in $\mathcal{D}$ by a corresponding application of the rule ARROW/FIX, more precisely, simply rename the rule name label attached to applications of ARROW from “ARROW” to “ARROW/FIX”.

(ii) Replace each application ι_R of a rule $R \in \{\mathrm{REN}, (\mu\!-\!\bot)^{\bot\mathrm{der}}_{l/r}, \mathrm{FOLD}_{l/r}\}$ by a respective derivation context $\mathcal{DC}^{(\iota_R)}_{\mathrm{mim}} \in \mathcal{D}er\mathcal{C}txt_1(\mathbf{HB}^{=})$ as described, for one case, in (a), i.e. replace each subderivation $\mathcal{D}_0$ ending with an application ι_R of such a rule and of the form

$$\dfrac{\begin{array}{c}\mathcal{D}_{01}\\ \tau_{01} = \sigma_{01}\end{array}}{\tau_0 = \sigma_0}\,R \quad \text{by a subderivation } \tilde{\mathcal{D}}_0 \text{ of the form} \quad \begin{array}{c}\mathcal{D}_{01}\\ [\tau_{01} = \sigma_{01}]_1\\ \mathcal{DC}^{(\iota_R)}_{\mathrm{mim}}\\ \tau_0 = \sigma_0\end{array}$$

accordingly (for all rules $R \in \{\mathrm{REN}, (\mu\!-\!\bot)^{\bot\mathrm{der}}_{l/r}, \mathrm{FOLD}_{l/r}\}$ and for all $\tau_0, \tau_{01}, \sigma_0, \sigma_{01} \in \mu Tp$); we notice hereby that $\mathcal{D}_0$ and $\tilde{\mathcal{D}}_0$ always contain the same open assumptions (if any) because the derivation contexts $\mathcal{DC}^{(\iota_R)}_{\mathrm{mim}}$ that are chosen here (to be of forms as detailed in (a)) do not contain applications of ARROW/FIX, the only rule of $\mathbf{HB}^{=}$ that enables applications at which open assumptions are discharged.

□

It is however not equally easy to give a proof-theoretic transformation from derivations in the system $\mathbf{HB}^{=}$ into derivations in the system $\mathbf{HB}^{=}_{\mathbf{0}}$. This is due to the presence of the rules SYMM and TRANS in $\mathbf{HB}^{=}$, for which it turns out that they are admissible, but not derivable rules of $\mathbf{HB}^{=}_{\mathbf{0}}$.

Still, the equivalence of the systems $\mathbf{HB}^{=}$ and $\mathbf{HB}^{=}_{\mathbf{0}}$ can also be established in a not proof-theoretic way by showing the soundness and completeness of $\mathbf{HB}^{=}_{\mathbf{0}}$ with respect to recursive type equality $=_\mu$ directly, i.e. without showing that derivations in $\mathbf{HB}^{=}$ (without open assumptions) can be transformed into mimicking derivations (without open assumptions) in $\mathbf{HB}^{=}_{\mathbf{0}}$ (a transformation in the opposite direction has been stated by Lemma 5.1.19 above). Actually, the soundness of $\mathbf{HB}^{=}_{\mathbf{0}}$ with respect to $=_\mu$ follows from Lemma 5.1.19 and from the soundness of $\mathbf{HB}^{=}$ with respect to $=_\mu$. And for the equivalence of the systems $\mathbf{HB}^{=}$ and $\mathbf{HB}^{=}_{\mathbf{0}}$ it would suffice to show, in view of the soundness of $\mathbf{HB}^{=}$ with respect to $=_\mu$, the completeness of $\mathbf{HB}^{=}_{\mathbf{0}}$ with respect to $=_\mu$.

A completeness theorem for the system $\mathbf{HB}^{=}_{\mathbf{0}}$ can be proven analogously as for $\mathbf{HB}^{=}$ in the way that is done in [BrHe98]. Interestingly, it turns out that, for all $\tau, \sigma \in \mu Tp$ with $\tau =_\mu \sigma$, the derivations $\mathcal{D}_{(\tau,\sigma)}$ in $\mathbf{HB}^{=}$ which are found effectively by the procedure underlying the completeness-proof in [BrHe98] correspond in a straightforward way[11] to derivations in the system $\mathbf{HB}^{=}_{\mathbf{0}}$. Therefore it can be argued that Brandt and Henglein have not only shown soundness and completeness of their system, which we have formalized as the system $\mathbf{HB}^{=}$ here, but they have implicitly also shown the somewhat *stronger* statement of the soundness and completeness with respect to $=_\mu$ of the TRANS- and SYMM-free 'kernel' of $\mathbf{HB}^{=}$, the system $\mathbf{HB}^{=}_{\mathbf{0}}$ in our formalization.

[11] Cf. the more specific remark in the proof sketch for Theorem 5.1.20 below.

Theorem 5.1.20 (Soundness and completeness of $\mathbf{HB}^{=}_{\mathbf{0}}$ w.r.t. $=_{\mu}$). *The variant system $\mathbf{HB}^{=}_{\mathbf{0}}$ of the system $\mathbf{HB}^{=}$ is sound and complete with respect to strong recursive type equivalence $=_{\mu}$, i.e. for all $\tau, \sigma \in \mu Tp$ it holds:*

$$\vdash_{\mathbf{HB}^{=}_{\mathbf{0}}} \tau = \sigma \quad \Longleftrightarrow \quad \tau =_{\mu} \sigma\,. \tag{5.9}$$

Sketch of Proof. Both the 'soundness-direction' and the 'completeness-direction' in the equivalence (5.9) can be proven in an entirely analogous way as was outlined for $\mathbf{HB}^{=}$ in the proof sketch for Theorem 5.1.12.

In particular it is the case, that the derivation $\mathcal{D}_{(\tau,\sigma)}$, which the completeness-proof in [BrHe98] produces effectively for arbitrary $\tau, \sigma \in \mu Tp$ with $\tau =_{\mu} \sigma$, can easily be rewritten as a proof $\mathcal{D}'_{(\tau,\sigma)}$ in $\mathbf{HB}^{=}_{\mathbf{0}}$; this is because the rules TRANS and SYMM do occur in the derivation $\mathcal{D}_{(\tau,\sigma)}$ for arbitrary $\tau, \sigma \in \mu Tp$ only immediately after axioms and can therefore be 'absorbed into' respective applications of the rules REN, $\mathrm{FOLD}_{l/r}$ or $(\mu{-}\bot)^{\bot\mathrm{der}}_{l/r}$ in $\mathbf{HB}^{=}_{\mathbf{0}}$. □

5.2 Proof Systems for Consistency-Checking with Respect to Recursive Type Equality

Now we turn our attention to a different kind of proof systems for recursive type equality $=_{\mu}$. These so called *syntactic-matching systems* are not axiomatizations of $=_{\mu}$, but are intended to provide a basis for *consistency-checking* with respect to $=_{\mu}$ for arbitrary given equations between recursive types. By this we mean: checking with respect to such a proof system whether a given equation between recursive types, or a considered set of equations between recursive types, can be added to this system *consistently*.

The idea for proof systems fit for consistency-checking by syntactic-matching originates, to our knowledge, from the paper [ArKl95] by Ariola and Klop and a proof system given there concerned with the notion of bisimilarity of cyclic term graphs. The system introduced in [ArKl95, see Table 1 on p. 223] is based on equational logic and enables checks for the consistency with respect to bisimulation equivalence of arbitrary equations between equational specifications of cyclic term graphs. The basic syntactic-matching system $\mathbf{AK}^{=}$ defined below is essentially a reformulation of the system studied by Ariola and Klop for a system with equations between recursive types as its formulas that bears upon the recursive type equality relation.

The usefulness of the systems defined in this section depends on a notion of "consistency", and dually on a notion of "inconsistency", of an equation between recursive types relative to a proof system. For defining these notions we have to stipulate when a proof system is actually inconsistent with respect to the recursive type equality relation $=_{\mu}$. And for this purpose we need to define first what "contradictions with respect to $=_{\mu}$" are.

Definition 5.2.1 (Contradictions with respect to $=_\mu$). An equation between recursive types $\tau = \sigma$ is called a *contradiction with respect to* $=_\mu$ if and only if τ and σ have different leading symbols, i.e. iff $\mathcal{L}'(\tau) \neq \mathcal{L}'(\sigma)$. We denote by $Ctrd_{=_\mu}$ the set of all equations between recursive types that are contradictions with respect to $=_\mu$. ⊠

Thus a contradiction $\tau = \sigma$ with respect to $=_\mu$ is the special case of an equation between recursive types for which it is very obvious that the recursive types τ and σ on its left- and on its right-hand side are not strongly equivalent: due to the definition of $\mathcal{L}'(\tau)$ and $\mathcal{L}'(\sigma)$, in this situation the tree unfoldings **Tree**(τ) and **Tree**(σ) of τ and of σ differ already by the symbols that respectively label the root of **Tree**(τ) and the root of **Tree**(σ).

Example 5.2.2 (Contradictions with respect to $=_\mu$). The following equations between recursive types are contradictions with respect to $=_\mu$:

$$\bot = \top\,, \quad \top = \alpha\,, \quad \alpha = \beta \ \ (\text{if } \alpha \not\equiv \beta)\,, \quad \mu\alpha_1 \ldots \alpha_n.\,\alpha_1 = \top\,,$$
$$\alpha = \mu\beta.\,(\tau_1 \to \tau_2)\,, \quad \tau_1 \to \tau_2 = \mu\alpha_1\alpha_2.\,\alpha_2\,, \quad \mu\alpha_1\alpha_2.\,\top = \mu\alpha.\,(\tau_1 \to \tau_2)$$

(with arbitrary $n \in \omega\backslash\{0\}$, $\alpha, \beta, \alpha_1, \alpha_2, \ldots, \alpha_n \in TVar$ and $\tau_1, \tau_2, \rho_1, \rho_2 \in \mu Tp$). But the following equations between recursive types are *not* contradictions with respect to $=_\mu$:

$$\mu\alpha.\,\alpha = \bot\,, \quad \tau_1 \to \tau_2 = \mu\alpha_1 \ldots \alpha_n.\,(\rho_1 \to \rho_2)\,,$$
$$\mu\alpha_1 \ldots \alpha_n.\,\alpha = \alpha \quad (\text{if } \alpha \not\equiv \alpha_1, \ldots, \alpha_n)$$

(for all $n \in \omega\backslash\{0\}$, $\alpha, \alpha_1, \alpha_2, \ldots, \alpha_n \in TVar$ and $\tau_1, \tau_2, \rho_1, \rho_2 \in \mu Tp$). ■

When using a notation like $\tau = \sigma, \Gamma \vdash_{\mathcal{S}} \chi_1 = \chi_2$ in the following, for some recursive types $\tau, \sigma, \chi_1, \chi_2$, a set of formulas $\Gamma \subseteq \mu Tp\text{–}Eq$ and a formal proof system $\mathcal{S}$ with $\mu Tp\text{–}Eq$ as its formulas, then we will mean that the formula $\chi_1 = \chi_2$ is derivable in the system $\mathcal{S}$ by a formal proof (a formal derivation) $\mathcal{D}$, in which each of the formulas in Γ and the formula $\tau = \sigma$ may be used *an arbitrary number of times* as an assumption (as a hypotheses); but all assumptions in $\mathcal{D}$ must be among the formulas $\tau = \sigma$ and those in the set Γ. We will also use the verbal equivalent " $\chi_1 = \chi_2$ is derivable in $\mathcal{S}$ from the assumption $\tau = \sigma$ and from assumptions in Γ" for the expression $\tau = \sigma,\ \Gamma \vdash_{\mathcal{S}} \chi_1 = \chi_2$.

Relying on the notion "contradiction with respect to $=_\mu$", we are now able to define the notions of consistency and inconsistency with respect to $=_\mu$ of a proof system, and of consistency and inconsistency relative to a proof system of equations and of sets of equations between recursive types.

Definition 5.2.3 (Consistency and inconsistency with respect to $=_\mu$ of proof systems, of equations, and of sets of equations betw. rec. types). Let $\mathcal{S}$ be a proof system whose formulas are precisely the equations between recursive types, i.e. the elements of $\mu Tp\text{–}Eq$.

(i) The system $\mathcal{S}$ is *inconsistent with respect to* $=_\mu$ if and only if $\mathcal{S}$ possesses a contradiction with respect to $=_\mu$ as a theorem. Dually, $\mathcal{S}$ is *consistent with respect to* $=_\mu$ if and only if no contradiction with respect to $=_\mu$ is derivable in $\mathcal{S}$.

In the following we will always drop the phrase "with respect to $=_\mu$" from an assertion "$\mathcal{S}$ is (in-)consistent with respect to $=_\mu$" and will always use "$\mathcal{S}$ is *(in-)consistent*" instead. No confusion will arise from this because no other notions of consistency or inconsistency for proof systems involving recursive types will be considered.

(ii) Let $\tau, \sigma \in \mu Tp$ be recursive types. We say that the equation $\tau = \sigma$ is *inconsistent relative to* $\mathcal{S}$, or *inconsistent with* $\mathcal{S}$, or, still shorter, $\mathcal{S}$*-inconsistent* if and only if the extension by enlargement $\mathcal{S}+\{\tau = \sigma\}$ of $\mathcal{S}$ is inconsistent; otherwise $\tau = \sigma$ is called *consistent relative to* $\mathcal{S}$, *consistent with* $\mathcal{S}$, or $\mathcal{S}$*-consistent.*

(iii) Let $\Gamma \subseteq \mu Tp\text{–}Eq$ be a set of equations between recursive types. Γ is called $\mathcal{S}$*-inconsistent relative to* $=_\mu$, $\mathcal{S}$*-inconsistent with* $=_\mu$, or $\mathcal{S}$*-inconsistent* if and only if the extension by enlargement $\mathcal{S}+\Gamma$ of $\mathcal{S}$ is inconsistent; otherwise Γ is called *consistent relative to* $\mathcal{S}$, *consistent with* $\mathcal{S}$, or $\mathcal{S}$*-consistent.*

⊠

Expanding, for example, the definition of $\mathcal{S}$-consistent equation between recursive types for a given proof system $\mathcal{S}$ with $\mu Tp\text{–}Eq$ as its set of formulas, we find that it holds for all $\tau, \sigma \in \mu Tp$:

$$\tau = \sigma \text{ is } \mathcal{S}\text{-consistent} \iff \begin{array}{l} \text{For all } \chi_1, \chi_2 \in \mu Tp: \\ \tau = \sigma \vdash_{\mathcal{S}} \chi_1 = \chi_2 \implies \mathcal{L}'(\chi_1) = \mathcal{L}'(\chi_2)\ . \end{array}$$

Now we give the definition of the 'syntactic-matching' proof system $\mathbf{AK}^=$ that differs from the axiom systems $\mathbf{AC}^=$, $\mathbf{AC}^=_*$, $\mathbf{HB}^=$, and $\mathbf{HB}^=_\mathbf{0}$ by the following noteworthy aspect: whereas $\mathbf{AC}^=$, $\mathbf{AC}^=_*$, $\mathbf{HB}^=$, and $\mathbf{HB}^=_\mathbf{0}$ contain either or both of the *composition rules* ARROW and ARROW/FIX, the system $\mathbf{AK}^=$ possesses a *decomposition rule* DECOMP.

Definition 5.2.4 (The 'syntactic-matching' proof system $\mathbf{AK}^=$ for $=_\mu$). The (pure) Hilbert-style proof system $\mathbf{AK}^=$ has as its *formulas* precisely the equations between recursive types in $\mu Tp\text{–}Eq$. It has the same *axioms* as the system $\mathbf{HB}^=$, namely the formulas of the schemes (REFL), (REN), $(\mu - \bot)'$ and (FOLD/UNFOLD). And as its *inference rules* it contains precisely the rules SYMM and TRANS as well as the *decomposition rule* DECOMP that are, together with the axioms, listed and schematically defined in Figure 5.6. ⊠

In the following example, we show for a particular example of an equation between recursive types that it is not consistent with $\mathbf{AK}^=$. We demonstrate this by giving a derivation in $\mathbf{AK}^=_\mathbf{0}$ that exhibits the particular behaviour of applications of the decomposition rule DECOMP in $\mathbf{AK}^=$.

Figure 5.6: The ‘syntactic-matching’ proof system $\mathbf{AK}^{=}$ fit for checking the consistency w.r.t. recursive type equality $=_\mu$ relative to this system of arbitrary equations between recursive types. This system is similar to one that was introduced by Ariola and Klop with respect to bisimulation equivalence on equational specifications of cyclic term graphs.

The *axioms* of $\mathbf{AK}^{=}$:

$$\text{(REFL)}\ \ \overline{\tau = \tau} \qquad\qquad \text{(REN)}\ \ \overline{\tau = \tau'}\ \ (\text{if } \tau \equiv_{\text{ren}} \tau')$$

$$(\mu - \bot)'\ \ \overline{\mu\alpha\alpha_1 \ldots \alpha_n.\, \alpha = \bot} \qquad \text{(FOLD/UNFOLD)}\ \ \overline{\mu\alpha.\, \tau = \tau[\mu\alpha.\, \tau/\alpha]}$$

The *inference rules* of $\mathbf{AK}^{=}$:

$$\frac{\sigma = \tau}{\tau = \sigma}\,\text{SYMM} \qquad\qquad \frac{\tau = \rho \qquad \rho = \sigma}{\tau = \sigma}\,\text{TRANS}$$

$$\frac{\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2}{\tau_i = \sigma_i}\,\text{DECOMP}\ \ (i = 1, 2)$$

Example 5.2.5 (An $\mathbf{AK}^{=}$-inconsistent equation between recursive types). We consider the recursive types

$$\tau_1 \equiv \mu\alpha.\, (\alpha \to \bot) \qquad \text{and} \qquad \tilde{\sigma}_1 \equiv \mu\beta.\, ((\beta \to \gamma) \to \bot) \quad (\text{where } \gamma \not\equiv \beta)$$

from Example 3.6.2, where we have also convinced ourselves that τ and $\tilde{\sigma}_1$ are not strongly equivalent. We are now going to demonstrate that the equation between recursive types $\tau_1 = \tilde{\sigma}_1$ is not consistent relative to the system $\mathbf{AK}^{=}$. For this, we consider the derivation $\mathcal{D}$ in $\mathbf{AK}^{=}$ of the form

$$\dfrac{\dfrac{\dfrac{\text{(FOLD/UNFOLD)}}{\tau_1 = \tau_1 \to \bot}}{\tau_1 \to \bot = \tau_1}\,\text{SYMM} \qquad \dfrac{\dfrac{\text{TRANS}\,\dfrac{\text{SYMM}\,\dfrac{\dfrac{\text{(FOLD/UNFOLD)}}{\tau_1 = \tau_1 \to \bot}}{\tau_1 \to \bot = \tau_1} \qquad \overset{\text{(Assm)}}{\tau_1 = \tilde{\sigma}_1}}{\tau_1 \to \bot = \tilde{\sigma}_1} \qquad \dfrac{\text{(FOLD/UNFOLD)}}{\tilde{\sigma}_1 = (\tilde{\sigma}_1 \to \gamma) \to \bot}}{\tau_1 \to \bot = (\tilde{\sigma}_1 \to \gamma) \to \bot}\,\text{TRANS}}{\tau_1 = \tilde{\sigma}_1 \to \gamma}\,\text{DECOMP}}{\dfrac{\tau_1 \to \bot = \tilde{\sigma}_1 \to \gamma}{\bot = \gamma}\,\text{DECOMP}}\,\text{TRANS}$$

Contradiction!

that contains $\tau_1 = \tilde{\sigma}_1$ as its (single) assumption and that has $\bot = \gamma$, a contradiction with respect to $=_\mu$, as its conclusion. Hence, according to the stipulations in Definition 5.2.3, the equation $\tau_1 = \tilde{\sigma}_1$ is *not* consistent relative to the system $\mathbf{AK}^{=}$; or $\mathbf{AK}^{=}$-inconsistent. ■

By using the notions defined above, we are eventually able to formulate the question how the notion of consistency relative to the system $\mathbf{AK^=}$ of an equation between recursive types corresponds to strong recursive type equivalence $=_\mu$. More formally, we can put the question whether at all, and if so then how, the binary relation $\mathrm{Cons}_{\mathbf{AK^=}}(\cdot,\cdot)$ on μTp that is defined by stipulating, for all $\tau,\sigma\in\mu Tp$,

$$\mathrm{Cons}_{\mathbf{AK^=}}(\tau,\sigma) \quad \Longleftrightarrow_{\mathrm{def}} \quad \tau=\sigma \text{ is } \mathbf{AK^=}\text{-consistent} \tag{5.10}$$

does in fact relate to $=_\mu$. The following theorem states that these two relations on μTp are actually equal, and in this way formulates a different kind of soundness and completeness result than the ones we have encountered for the systems $\mathbf{AC^=}$, $\mathbf{HB^=}$, and $\mathbf{HB_0^=}$. It is for reasons that will be discussed in Remark 5.2.7, (b), below, that we do not call this statement a “soundness and completeness theorem”.

Theorem 5.2.6 (Ariola, Klop, ’95: Correspondence between consistency with $\mathbf{AK^=}$ and recursive type equality). *The relation $\mathit{Cons}_{\mathbf{AK^=}}(\cdot,\cdot)$ on μTp coincides with recursive type equality $=_\mu$, i.e. for all $\tau,\sigma\in\mu Tp$ it holds:*

$$\tau=\sigma \textit{ is } \mathbf{AK^=}\textit{-consistent} \quad\Longleftrightarrow\quad \tau=_\mu\sigma\ . \tag{5.11}$$

Sketch of Proof. It suffices to prove (5.11) for all $\tau,\sigma\in\mu Tp$. We shall only give a sketch for the proof of both directions of this equivalence.

“$\Rightarrow$”: The essential tool in our proof of this direction in (5.11) is the following general statement about the existence for all $\tau,\sigma\in\mu Tp$ of such derivations $\mathcal{D}$ in $\mathbf{AK^=}$ from the assumption $\tau=\sigma$ that correspond to given access paths p in the common part of the tree unfoldings $\mathbf{Tree}(\tau)$ and $\mathbf{Tree}(\sigma)$ of τ and σ:

$$\left.\begin{array}{l}(\forall\,\tau,\sigma\in\mu Tp)\ (\forall\,p\in Acc(\tau)\cap Acc(\sigma)\,)\\ \qquad(\exists\,\chi_1,\chi_2\in\mu Tp)\\ \qquad\qquad\big(\,\exists\,\mathcal{D}\ \text{ in } \mathbf{AK^=} \text{ with the single assumption}\\ \qquad\qquad\qquad\qquad \tau=\sigma\ \text{ and with conclusion }\ \chi_1=\chi_2\,\big)\\ \qquad\qquad\quad\big[\,\mathbf{Tree}(\tau)|_p=\mathbf{Tree}(\chi_1)\ \ \&\ \ \mathbf{Tree}(\sigma)|_p=\mathbf{Tree}(\chi_2)\,\big]\ .\end{array}\right\} \tag{5.12}$$

(We recall that $Acc(\rho)=_{\mathrm{def}}\mathrm{dom}(\mathbf{Tree}(\rho))$ for all $\rho\in\mu Tp$). This can be shown in a very straightforward way by induction on access paths p.

We now let $\tau,\sigma\in\mu Tp$ be arbitrary such that $\tau=\sigma$ is $\mathbf{AK^=}$-consistent. We want to show that τ and σ have the same tree unfolding, i.e. that $\tau=_\mu\sigma$.

We *assume* that $\mathbf{Tree}(\tau)\neq\mathbf{Tree}(\sigma)$ (and will derive from this a contradiction to the $\mathbf{AK^=}$-consistency of $\tau=\sigma$). It follows that $p\in Acc(\tau)\cap Acc(\sigma)\subseteq$ $\subseteq\{1,2\}^*$ exists such that $\mathbf{Tree}(\tau)(p)\neq\mathbf{Tree}(\sigma)(p)$; we choose one such path p. From (5.12) we conclude the existence of a derivation $\mathcal{D}$ in $\mathbf{AK^=}$ with one single assumption $\tau=\sigma$ and with conclusion $\chi_1=\chi_2$ such that $\mathbf{Tree}(\tau)|_p=\mathbf{Tree}(\chi_1)$ and $\mathbf{Tree}(\sigma)|_p=\mathbf{Tree}(\chi_2)$. But this implies

$$\begin{aligned}\mathcal{L}'(\chi_1)=\mathbf{Tree}(\chi_1)(\epsilon)=\mathbf{Tree}(\tau)|_p(\epsilon)=\mathbf{Tree}(\tau)(p)\neq&\\ \neq\mathbf{Tree}(\sigma)(p)=\mathbf{Tree}(\sigma)|_p(\epsilon)=\mathbf{Tree}(\chi_2)(\epsilon)=\mathcal{L}'(\chi_2)&\end{aligned}$$

and hence $\mathcal{L}'(\chi_1) \neq \mathcal{L}'(\chi_2)$. It follows that $\tau = \sigma \vdash_{\mathbf{AK}^=} \chi_1 = \chi_2$ holds for some $\chi_1, \chi_2 \in \mu Tp$ with the property $\mathcal{L}'(\chi_1) \neq \mathcal{L}'(\chi_2)$. But this would mean that $\tau = \sigma$ were in fact not $\mathbf{AK}^=$-consistent, in contradiction with our choice of τ and σ.

We conclude that our assumption was false. Therefore $\mathbf{Tree}(\tau) = \mathbf{Tree}(\sigma)$ and $\tau =_\mu \sigma$ must hold.

"$\Leftarrow$": For the proof of this direction in (5.11) we use an auxiliary statement of the following form:

$$\left.\begin{array}{l} \text{Suppose that} \quad \begin{array}{c} \{\,[\tau_i = \sigma_i]\}_{i=1,\ldots,n} \\ \mathcal{D} \\ \chi_1 = \chi_2 \end{array} \quad \text{is a derivation in } \mathbf{AK}^= \\ \text{with conclusion } \chi_1 = \chi_2\text{, for some } \chi_1, \chi_2 \in \mu Tp\text{, where the as-} \\ \text{sumptions in leaves at the top of } \mathcal{D} \text{ are precisely those that be-} \\ \text{long to one of the } n \text{ marked assumption classes } [\tau_i = \sigma_i]\text{, with} \\ \text{some } \tau_i, \sigma_i \in \mu Tp,\ i \in \{1, \ldots, n\} \text{ and with } n \in \omega\text{. Suppose fur-} \\ \text{ther that also } \mathbf{Tree}(\tau_i) = \mathbf{Tree}(\sigma_i) \text{ holds for all } 1 \leq i \leq n. \\ \text{Then it follows that also } \mathbf{Tree}(\chi_1) = \mathbf{Tree}(\chi_2) \text{ holds.} \end{array}\right\} \quad (5.13)$$

This can be shown in a straightforward way by induction on the depth $|\mathcal{D}|$ of the $\mathbf{AK}^=$-derivation $\mathcal{D}$.

For the proof of "$\Leftarrow$" in (5.11) we now let $\tau, \sigma \in \mu Tp$ with $\tau =_\mu \sigma$ and hence with $\mathbf{Tree}(\tau) = \mathbf{Tree}(\sigma)$ be arbitrary. We have to show that $\tau = \sigma$ is $\mathbf{AK}^=$-consistent.

For this let $\mathcal{D}$ be an arbitrary derivation in $\mathbf{AK}^=$ with conclusion $\chi_1 = \chi_2$ and with only one class $[\tau = \sigma]$ of assumptions. We have to demonstrate that $\mathcal{L}'(\chi_1) = \mathcal{L}'(\chi_2)$. From (5.13) we can conclude for our situation that $\mathbf{Tree}(\chi_1) = \mathbf{Tree}(\chi_2)$. This clearly implies $\mathcal{L}'(\chi_1) = \mathcal{L}'(\chi_2)$.

Hence we have shown that $\tau = \sigma$ is consistent with $\mathbf{AK}^=$.

□

Remark 5.2.7. (a) Theorem 5.2.6 is similar to, and in fact was motivated by, the result formulated in Proposition 3.24 on page 224 in [ArKl95] (and in Proposition 3.37 on page 26 in the report version of [ArKl95]), where the authors consider the notion of bisimulation equivalence $\underline{\leftrightarrow}$ on cyclic term graphs and give a syntactic-matching system based on equational logic that we shall denote here by $\mathbf{AK}^{\underline{\leftrightarrow}}$. The formulas of the system $\mathbf{AK}^{\underline{\leftrightarrow}}$ are equations between equational representations of cyclic term graphs as finite recursion systems (finite sets of equations of a particular form). $\mathbf{AK}^{\underline{\leftrightarrow}}$ has been designed for considering the question: which formulas can be added to this system consistently, i.e. such that no "contradictions" become derivable; hereby an equation $\mathcal{R}_1 = \mathcal{R}_2$ between recursion systems $\mathcal{R}_1$ and $\mathcal{R}_2$ is called a "contradiction" if and only if $\mathcal{R}_1$ and $\mathcal{R}_2$ represent cyclic term graphs having different root-symbols.

The correspondence result for $\mathbf{AK}^{\leftrightarrows}$-consistency presented in [ArKl95] (formulated in the propositions mentioned above) asserts the following: two cyclic term graphs g_1 and g_2 denoted by respective recursion systems $\mathcal{R}_{g_1}$ and $\mathcal{R}_{g_2}$ are bisimilar (i.e. $g_1 \leftrightarrows g_2$ holds) if and only if no contradiction is derivable in $\mathbf{AK}^{\leftrightarrows}$ from the set $\mathcal{R}_{g_1} \cup \mathcal{R}_{g_2} \cup \{\alpha = \beta\}$, where α and β are the left-hand sides of the head-equations in $\mathcal{R}_{g_1}$ and $\mathcal{R}_{g_2}$ (i.e. those equations in $\mathcal{R}_{g_1}$ and $\mathcal{R}_{g_2}$ that specify g_1 and g_2 just below their roots and that contain the root-symbols of g_1 and g_2), respectively.[12]

(b) It is quite tempting to view Theorem 5.2.6 as a "soundness and completeness statement" for $\mathbf{AK}^{=}$: that is to say, more precisely, a soundness and completeness theorem for $\mathbf{AK}^{=}$-consistency with respect to recursive type equality. In particular, the implications "$\Rightarrow$" and "$\Leftarrow$" in (5.11) could be considered to justify, respectively, the "soundness" and the "completeness" with respect to $=_\mu$ of the notion "$\mathbf{AK}^{=}$-consistency". However, for reasons to be explained below, we have avoided to use the terms "soundness" and "completeness" in connection with Theorem 5.2.6.

Soundness and completeness theorems in formal logic tend to have the following form: for the formulas of a given language L, a syntactically defined, and typically "positively calculable",[13] notion $\vdash_{\mathcal{S}}$ of *formal deducibility*, or *provability* involving formulas of L in a formal system $\mathcal{S}$ is stated to coincide with a semantically defined notion $\vDash$ of *truth* of formulas of L in a certain formal semantics. The soundness part of such a theorem consists in the inclusion $\vdash_{\mathcal{S}} \subseteq \vDash$, i.e. it asserts that derivable formulas or statements of derivable consequence between formulas in $\mathcal{S}$ are true in the considered semantics; and the completeness part is the statement $\vDash \subseteq \vdash_{\mathcal{S}}$, i.e. the assertion that true formulas or consequence-statements are in fact formally derivable in $\mathcal{S}$. Usually the soundness part of such a theorem can be shown by induction on derivations in $\mathcal{S}$; contrasting with this, a proof of the completeness part is typically much more difficult and frequently requires rather sophisticated methods.

The term "positively calculable notion" used above refers to the following definition of formal notions that can be viewed as predicates (this informal definition is oriented at [Shoe67, p.121], but slightly rephrased): a predicate P for a set O of objects is said to be *positively calculable* if and only if O consists of 'concrete' objects and if there exists an *effective positive test* for checking, for all $o \in O$, whether $P(o)$ holds. Hereby such an effective positive test is an effective method M with the property that, for all $o \in O$, if $P(o)$ holds, then this can be verified in a finite number of steps by applying M to o; if $P(o)$ does not hold, then M either says so or gives no answer. [Shoe67] points out the connection that holds between positively calculable and 'recursively enumerable' predicates: if Church's thesis is assumed, a predicate is positively calculable if and only if it is 'recursively enumerable'.

Opposite to the typical situation for a notion of provability in a formal system, $\mathbf{AK}^{=}$-consistency of an equation between recursive types cannot be recognized im-

[12] For this statement to be correct, it must be assumed that the recursion (or 'bound') variables in the recursion systems $\mathcal{R}_{g_1}$ and $\mathcal{R}_{g_2}$ are mutually distinct; this caveat is missing in [ArKl95].

[13] See the informal definition of "positively calculable" in the following paragraph.

mediately as a positively calculable notion on the basis of its definition. This is because the definition of an equation $\tau = \sigma$ between recursive types to be consistent with respect to $\mathbf{AK}^{=}$ amounts, when expanded, to the stipulation

$$\begin{aligned} &\tau = \sigma \text{ is } \mathbf{AK}^{=}\text{-consistent} \iff \\ &\qquad\iff \neg\,(\exists\,\mathcal{D} \in \mathcal{D}er(\mathbf{AK}^{=}))\,\big[\,\mathrm{set}(\mathsf{assm}(\mathcal{D})) = \{\tau = \sigma\} \;\;\& \\ &\qquad\qquad\qquad\qquad \& \;\; \mathsf{concl}(\mathcal{D}) \in Ctrd_{=_\mu}\,\big] \end{aligned}$$

(for all $\tau, \sigma \in \mu Tp$). Viewed purely on itself, this stipulation does not suggest any effective positive test for determining for an equation $\tau = \sigma$ between recursive types whether it is consistent with respect to $\mathbf{AK}^{=}$. There might simply not exist an effective method to ascertain of an equation $\tau = \sigma$ that there does *not* exist a derivation with $\tau = \sigma$ which leads to a contradiction. But the situation is different for the notion of $\mathbf{AK}^{=}$-inconsistency of an equation $\tau = \sigma$, the definition of which, when expanded, has the form

$$\begin{aligned} &\tau = \sigma \text{ is } \mathbf{AK}^{=}\text{-inconsistent} \iff \\ &\iff (\exists\,\mathcal{D} \in \mathcal{D}er(\mathbf{AK}^{=}))\,\big[\,\mathrm{set}(\mathsf{assm}(\mathcal{D})) = \{\tau = \sigma\} \;\;\&\;\; \mathsf{concl}(\mathcal{D}) \in Ctrd_{=_\mu}\,\big] \end{aligned}$$

(for all $\tau, \sigma \in \mu Tp$). This implies that $\mathbf{AK}^{=}$-inconsistency is a positively calculable notion because there exists an effective positive test for making sure that an equation $\tau = \sigma$ between recursive types is inconsistent with respect to $\mathbf{AK}^{=}$: successively generate all possible derivations in $\mathbf{AK}^{=}$ in which precisely the assumption $\tau = \sigma$ occurs, and check whether any of these derivations has a contradiction as its conclusion; if such a derivation is found, then clearly $\tau = \sigma$ is inconsistent with respect to $\mathbf{AK}^{=}$. (Quite obviously, this procedure can also serve as an "effective negative test" for $\mathbf{AK}^{=}$-consistency.)

Due to these considerations, the statement

$$(\forall \tau, \sigma \in \mu Tp)\,\big[\,\tau = \sigma \text{ is } \mathbf{AK}^{=}\text{-inconsistent} \iff \tau \neq_\mu \sigma\,\big]\,, \tag{5.14}$$

where $\neq_\mu$ is the complement of recursive type equality $=_\mu$ on the set $\mu Tp \times \mu Tp$, would be more in line with most soundness and completeness theorems in logic. It could justly be called a soundness and completeness theorem of $\mathbf{AK}^{=}$-inconsistency with respect to $\neq_\mu$ (with "soundness" being asserted by the implication "⇒" in (5.14), and "completeness" by "⇐"). It is clear that (5.14) follows from, and in fact is equivalent with, Theorem 5.2.6, because, for all $\tau, \sigma \in \mu Tp$, each of the implications within (5.14) is equivalent, via contraposition, with a respective one of the two implications in (5.11).

Taking this view seriously, and appreciating the link via contraposition between the implications in (5.14) and in (5.11), one is led to say that in (5.11) the implication "⇐" has the 'flavour' of a soundness, and "⇒" that of a completeness statement; this would also correspond with the fact that in proof of Theorem 5.2.6 the implication "⇐" is proved by using an assertion that is shown by induction on $\mathbf{AK}^{=}$-derivations, whereas the proof of "⇒" requires a little more insight into the 'unwinding semantics' for recursive types.

However, we have decided to avoid the terms "soundness" and "completeness" in connection with Theorem 5.2.6 altogether, primarily because using these terms for the implications in (5.11) in the way as just suggested would clash with the 'naive' manner, explained at the start of this remark, of assigning these terms to the two implications in (5.11). A further reason is that our first interest is obviously not in a 'negative' statement like (5.14), but in a result that describes the connection between the notion of consistency relative to $\mathbf{AK}^{=}$ and recursive type equality in a more direct way. Therefore we have decided to call Theorem 5.2.6 a "correspondence theorem" between $\mathbf{AK}^{=}$-consistency and the relation $=_\mu$.

By Theorem 5.2.6, those formulas of $\mathbf{AK}^{=}$ that can be added consistently *as single formulas* to the system $\mathbf{AK}^{=}$ are precisely the equations between recursive types $\tau = \sigma$ with strongly equivalent $\tau, \sigma \in \mu Tp$. The following easy generalization of this theorem provides the answer to the question which *sets* of formulas from μTp–Eq can be added consistently to the system $\mathbf{AK}^{=}$.

Theorem 5.2.8 (Generalized version of Theorem 5.2.6). *Let* $\Gamma \subseteq \mu Tp$–Eq *be an arbitrary set of equations between recursive types. Then the following holds:*

$$\Gamma \textit{ is consistent with } \mathbf{AK}^{=} \quad\Longleftrightarrow\quad (\forall\, \tau = \sigma \,\in\, \Gamma)\,\big[\,\tau =_\mu \sigma\,\big]\,. \tag{5.15}$$

Proof. The proof of the direction "$\Leftarrow$" in (5.15) is, as was the case for the proof for the direction "$\Leftarrow$" of (5.11) in Theorem 5.2.6, an application of the statement (5.13) in the proof sketch for Theorem 5.2.6.

The direction "$\Rightarrow$" in (5.15) follows from (the direction "$\Rightarrow$" of the equivalence (5.11) in) Theorem 5.2.6 by using the obvious fact that every formula that belongs to an $\mathbf{AK}^{=}$-consistent set of formulas in μTp–Eq is itself $\mathbf{AK}^{=}$-consistent. □

For the investigation in later chapters of proof-theoretic interconnections between systems that are described in this chapter, it will be important to have also a variant system $\mathbf{AK}^{=}_{\mathbf{0}}$ of $\mathbf{AK}^{=}$ available that is defined below. The situation is similar to the case of the system $\mathbf{HB}^{=}$ for which we have introduced the variant system $\mathbf{HB}^{=}_{\mathbf{0}}$ without symmetry and transitivity rules. Indeed, also the system $\mathbf{AK}^{=}_{\mathbf{0}}$ is motivated by the wish to build a system equivalent to $\mathbf{AK}^{=}$ without symmetry and transitivity rules that has stronger proof-theoretical properties. Again, the desire to isolate a SYMM- and TRANS-free 'kernel' of the syntactic-matching system $\mathbf{AK}^{=}$ is due to the fact that the possibility to use rules for symmetry and transitivity accounts for a considerable amount of complexity in the structure of possible derivations in $\mathbf{AK}^{=}$. In the absence of SYMM and TRANS, the operations "renaming of bound variables", "unfolding of a recursive type", and "equating a recursive type $\mu\alpha_1 \ldots \alpha_n.\, \alpha_1$ with $\bot$ " that are expressed by the axioms of $\mathbf{AK}^{=}$ have to be formalized by respective rules in $\mathbf{AK}^{=}_{\mathbf{0}}$. These rules allow to perform such operations on either or on both sides of the equation in their premises.

Definition 5.2.9 (The variant system $\mathbf{AK}^{=}_{\mathbf{0}}$ of the the "syntactic-matching" system $\mathbf{AK}^{=}$). The Hilbert-style proof system $\mathbf{AK}^{=}_{\mathbf{0}}$ is defined as follows. The *formulas* of the formal system $\mathbf{AK}^{=}_{\mathbf{0}}$ are the equations between recursive types in μTp–Eq. $\mathbf{AK}^{=}_{\mathbf{0}}$ does not have any *axioms* and its *inference rules* are precisely

Figure 5.7: A ‘normalized’ version $\mathbf{AK}^{=}_{\mathbf{0}}$ without symmetry and transitivity rules of the “syntactic-matching” system $\mathbf{AK}^{=}$ fit for checking the consistency with respect to rec. type equality $=_\mu$ of arbitrary given equations between recursive types.

The *inference rules* of $\mathbf{AK}^{=}_{\mathbf{0}}$:

$$\frac{\mu\alpha\alpha_1\ldots\alpha_n.\,\alpha=\sigma}{\bot=\sigma}\;(\mu{-}\bot)_l^{\mathrm{der}\bot}\qquad\qquad \frac{\tau=\mu\beta\beta_1\ldots\beta_n.\,\beta}{\tau=\bot}\;(\mu{-}\bot)_l^{\mathrm{der}\bot}$$

$$\frac{\mu\alpha.\,\tau=\sigma}{\tau[\mu\alpha.\,\tau/\alpha]=\sigma}\;\mathrm{UNFOLD}_l\qquad\qquad \frac{\tau=\mu\beta.\,\sigma}{\tau=\sigma[\mu\beta.\,\sigma/\beta]}\;\mathrm{UNFOLD}_r$$

$$\frac{\tau_1\to\tau_2=\sigma_1\to\sigma_2}{\tau_i=\sigma_i}\;\mathrm{DECOMP}\ (i=1,2)\qquad\qquad \frac{\tau=\sigma}{\tau'=\sigma'}\;\mathrm{REN}\ \ \begin{array}{l}(\text{if }\tau'\equiv_{\mathrm{ren}}\tau\\ \ \ \text{and }\sigma'\equiv_{\mathrm{ren}}\sigma)\end{array}$$

the rules REN, $(\mu{-}\bot)_{l/r}^{\mathrm{der}\bot}$, $\mathrm{UNFOLD}_{l/r}$ and DECOMP that are schematically defined in Figure 5.7. (In this sentence we have again ‘bundled’ respective “l̲eft”- and “r̲ight”-sided rules R_l and R_r to rules $R_{l/r}$ for $R\in\{\text{UNFOLD, }(\mu{-}\bot)^{\mathrm{der}\bot}\}$.) ☒

Similarly as the system $\mathbf{HB}^{=}_{\mathbf{0}}$ can be viewed as a ‘normalized’ version of the basic Brandt-Henglein system $\mathbf{HB}^{=}$, also the system $\mathbf{AK}^{=}_{\mathbf{0}}$ can be considered to be a ‘normalized’ version of $\mathbf{AK}^{=}$ that does not possess symmetry and transitivity rules. However, there is a difference: in general, derivations in $\mathbf{AK}^{=}$ cannot be ‘normalized’ with the outcome of a derivation in $\mathbf{AK}^{=}_{\mathbf{0}}$ that has the same conclusion and the same assumptions because, unlike this is the case for derivations in $\mathbf{AK}^{=}_{\mathbf{0}}$, more than one unproven assumption is able to occur in derivations in $\mathbf{AK}^{=}$. Nevertheless, we will see below that an analogous correspondence result with $=_\mu$ holds for $\mathbf{AK}^{=}_{\mathbf{0}}$-consistency as the one stated by Theorem 5.2.6 for $\mathbf{AK}^{=}$-consistency.

The reason why we have spoken of $\mathbf{AK}^{=}_{\mathbf{0}}$ as of a ‘normalized’ version of the system $\mathbf{AK}^{=}$ consists in the fact that, in contrast with $\mathbf{AK}^{=}$, the system $\mathbf{AK}^{=}_{\mathbf{0}}$ fulfills a subformula property similar to the subformula property SP_1 of Definition 5.1.16, for which we have seen that it holds for $\mathbf{HB}^{=}_{\mathbf{0}}$.

Definition 5.2.10 (The subformula property SP_2). Let $\mathcal{S}$ be a Hilbert-style or natural-deduction-style proof system with $\mu\mathit{Tp}{-}\mathit{Eq}$ as its set of formulas.

We say that a derivation $\mathcal{D}$ in $\mathcal{S}$ with possibly (open) assumptions *satisfies*, or *fulfills*, *the subformula property* SP_2 if and only if the conclusion of $\mathcal{D}$ is a ‘subformula’ of all formulas that occur in $\mathcal{D}$. And we say that $\mathcal{S}$ *has*, or *obeys*, *the subformula property* SP_2, or that SP_2 *holds in* $\mathcal{S}$ if and only if every derivation in $\mathcal{S}$ satisfies SP_2. ☒

Quite obviously, the subformula property SP_2 is the *inverse form* of the subformula property SP_1 defined in Definition 5.1.16: for derivations $\mathcal{D}$ of a proof system,

Figure 5.8: Illustration of the difference between what the *subformula properties* SP_1 and SP_2 demand respectively, for arbitrary derivations $\mathcal{D}$ in proof systems with set $\mu Tp\text{–}Eq$ of formulas, about the relationship of formulas occurring in $\mathcal{D}$ with the conclusion of $\mathcal{D}$.

derivation $\mathcal{D}$ with
occurring formula $\chi_1 = \chi_2$
and conclusion $\tau = \sigma$

$$\boxed{\begin{array}{c} \mathcal{D}_1 \\ (\chi_1 = \chi_2) \\ \mathcal{DC} \\ \tau = \sigma \end{array}}$$

$\mathcal{D}$ fulfills the subformula property SP_1
$\iff$ $\chi_1 \sqsubseteq' \tau$ & $\chi_2 \sqsubseteq' \sigma$
($\chi_1 = \chi_2$ is a ‘subformula’ of $\tau = \sigma$)

derivation $\mathcal{D}$ with
occurring formula $\chi_1 = \chi_2$
and conclusion $\tau = \sigma$

$$\boxed{\begin{array}{c} \mathcal{D}_1 \\ (\chi_1 = \chi_2) \\ \mathcal{DC} \\ \tau = \sigma \end{array}}$$

$\mathcal{D}$ fulfills the subformula property SP_2
$\iff$ $\tau \sqsubseteq' \chi_1$ & $\sigma \sqsubseteq' \chi_2$
($\tau = \sigma$ is a ‘subformula’ of $\chi_1 = \chi_2$)

fulfilledness of SP_2 requires the conclusion of $\mathcal{D}$ to be a ‘subformula’ of each formula occurring in $\mathcal{D}$, whereas fulfilledness of SP_1 demands, *vice versa*, that each formula occurring in $\mathcal{D}$ is a ‘subformula’ of the conclusion of $\mathcal{D}$; see also Figure 5.8 for an illustration of this difference.

It is straightforward to verify that SP_2 does not hold in $\mathbf{HB}_{\mathbf{0}}^{=}$ (and neither in any of the other axiom systems for $=_\mu$ that we have encountered earlier), and that SP_1 does not hold in $\mathbf{AK}_{\mathbf{0}}^{=}$. However, $\mathbf{AK}_{\mathbf{0}}^{=}$ obeys the subformula property SP_2.

Proposition 5.2.11. *The system* $\mathbf{AK}_{\mathbf{0}}^{=}$ *obeys the subformula property* SP_2. *And furthermore, for all* $\tau, \sigma \in \mu Tp$ *and derivations* $\mathcal{D}$ *in* $\mathbf{AK}_{\mathbf{0}}^{=}$ *with conclusion* $\tau = \sigma$ *the following holds: each formula* $\chi_1 = \chi_2$ *that occurs in* $\mathcal{D}$ *is a ‘subformula’ of the single assumption* $\tau = \sigma$ *of* $\mathcal{D}$.

Proof. The fact that the subformula property SP_2 holds in $\mathbf{AK}_{\mathbf{0}}^{=}$ can be demonstrated by a straightforward proof by induction on the depth $|\mathcal{D}|$ of a derivation $\mathcal{D}$ in $\mathbf{AK}_{\mathbf{0}}^{=}$. In the induction step the transitivity of the $\rightarrow_{\mathrm{roud}\perp}$-generated-subterm relation $\sqsubseteq'$ is used as well as the observation that if $\dfrac{\tilde{\tau}_0 = \tilde{\sigma}_0}{\tilde{\tau} = \tilde{\sigma}}\, R$ is an application of a rule R in $\mathbf{AK}_{\mathbf{0}}^{=}$, then $\tilde{\tau} \sqsubseteq' \tilde{\tau}_0$ and $\tilde{\sigma} \sqsubseteq' \tilde{\sigma}_0$ hold.

The second assertion of the proposition is an immediate consequence. To demonstrate this, let $\mathcal{D}$ be an arbitrary derivation in $\mathbf{AK}_{\mathbf{0}}^{=}$ and let us consider an arbitrary occurrence in $\mathcal{D}$ of the form $\chi_1 = \chi_2$, for some $\chi_1, \chi_2 \in \mu Tp$. Furthermore, let $\tau = \sigma$ be the single assumption of $\mathcal{D}$ and let $\mathcal{D}_0$ be the immediate subderivation of the considered occurrence of $\chi_1 = \chi_2$ in $\mathcal{D}_0$ (i.e. the subderivation of $\mathcal{D}$ that

Figure 5.9: Illustration of the difference between what the subformula properties SP_1 and SP_2 imply for derivations in $\mathbf{HB}_{\mathbf{0}}^{=}$ and derivations in $\mathbf{AK}_{\mathbf{0}}^{=}$, respectively: in a $\mathbf{HB}_{\mathbf{0}}^{=}$-derivation $\mathcal{D}$ each formula is a ‘subformula’ of the conclusion of $\mathcal{D}$; in an $\mathbf{AK}_{\mathbf{0}}^{=}$-derivation $\mathcal{D}$ each formula is a ‘subformula’ of the single assumption of $\mathcal{D}$.

$\mathbf{HB}_{\mathbf{0}}^{=}$-derivation $\mathcal{D}$
with the conclusion $\tau = \sigma$

$$\boxed{\begin{array}{c}\mathcal{D}_0\\ (\chi_1 = \chi_2)\\ \mathcal{DC}\\ \tau = \sigma\end{array}}$$

$\chi_1 \sqsubseteq' \tau \quad \& \quad \chi_2 \sqsubseteq' \sigma$
($\chi_1 = \chi_2$ is a ‘subformula’ of $\tau = \sigma$)

$\mathbf{AK}_{\mathbf{0}}^{=}$-derivation $\mathcal{D}$
from the assumption $\tau = \sigma$

$$\boxed{\begin{array}{c}(\tau = \sigma)\\ \mathcal{D}_1\\ (\chi_1 = \chi_2)\\ \mathcal{D}_2\end{array}}$$

$\chi_1 \sqsubseteq' \tau \quad \& \quad \chi_2 \sqsubseteq' \sigma$
($\chi_1 = \chi_2$ is a ‘subformula’ of $\tau = \sigma$)

ends in the considered occurrence of $\chi_1 = \chi_2$). Clearly $\mathcal{D}_0$ is an $\mathbf{AK}_{\mathbf{0}}^{=}$-derivation; it has conclusion $\chi_1 = \chi_2$ and the single assumption $\tau = \sigma$. Since the subformula property SP_2 is fulfilled for $\mathbf{AK}_{\mathbf{0}}^{=}$, for which we have hinted the proof above, it follows that $\chi_1 = \chi_2$ is a ‘subformula’ of $\tau = \chi$. □

For an illustration of the difference between what the subformula property SP_1 that holds in $\mathbf{HB}_{\mathbf{0}}^{=}$ and the subformula property SP_2 that holds in $\mathbf{AK}_{\mathbf{0}}^{=}$ entail for derivations in $\mathbf{HB}_{\mathbf{0}}^{=}$ and for derivations in $\mathbf{AK}_{\mathbf{0}}^{=}$, respectively, see Figure 5.9. Analogously as in Remark 5.1.18, where we have argued that the subformula property SP_1 does not hold in the system $\mathbf{HB}^{=}$ due to the presence there of the transitivity rule TRANS, it can be demonstrated that the subformula property SP_2 does not hold in the system $\mathbf{AK}^{=}$ which also contains TRANS.

We continue with an example in which we establish $\mathbf{AK}_{\mathbf{0}}^{=}$-inconsistency of the particular equation between recursive types for which we have demonstrated its inconsistency relative to $\mathbf{AK}^{=}$ in Example 5.2.5.

Example 5.2.12 ($\mathbf{AK}_{\mathbf{0}}^{=}$-inconsistency of the equation between recursive types used in Example 5.2.5). As in Example 5.2.5, we again consider the two recursive types $\tau_1 \equiv \mu\alpha.\,(\alpha \to \bot)$ and $\tilde{\sigma}_1 \equiv \mu\beta.\,((\beta \to \gamma) \to \bot)$ (where $\gamma \not\equiv \beta$) from Example 3.6.2, which are not strongly equivalent. It is now even easier to demonstrate that the equation $\tau_1 = \tilde{\sigma}_1$ is not $\mathbf{AK}_{\mathbf{0}}^{=}$-consistent than it was in Example 5.2.5 to show $\mathbf{AK}^{=}$-inconsistency of $\tau_1 = \tilde{\sigma}_1$. This is because the derivation $\tilde{\mathcal{D}}$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ of the simple form

$$\begin{array}{cl}
(\text{Assm}) & \\
\overbrace{\mu\alpha.\,(\alpha \to \bot)}^{\equiv\,\tau_1} = \overbrace{\mu\beta.\,((\beta \to \gamma) \to \bot)}^{\equiv\,\tilde{\sigma}_1} & \\
\hline\hline
\tau_1 \to \bot = (\tilde{\sigma}_1 \to \gamma) \to \bot & \text{UNFOLD}_{l/r} \\
\hline
\tau_1 = \tilde{\sigma}_1 \to \gamma & \text{DECOMP} \\
\hline
\tau_1 \to \bot = \tilde{\sigma}_1 \to \gamma & \text{UNFOLD}_{l} \\
\hline
\bot = \gamma & \text{DECOMP} \\
\textbf{Contradiction!} &
\end{array} \tag{5.16}$$

has the assumption $\tau_1 = \tilde{\sigma}_1$ and the conclusion $\bot = \gamma$, which is a contradiction with respect to $=_\mu$. Hence it follows again from Definition 5.2.3 that the equation $\tau_1 = \tilde{\sigma}_1$ is *not consistent* with the system $\mathbf{AK}_{\mathbf{0}}^{=}$. As an aside, we observe that the derivation $\mathcal{D}$ in Example 5.2.5 can be considered to be the 'translation' to $\mathbf{AK}^{=}$ of the derivation[14] $\tilde{\mathcal{D}}$ in $\mathbf{AK}_{\mathbf{0}}^{=}$. The transformation effected hereby is analogous to the transformation between $\mathbf{HB}_{\mathbf{0}}^{=}$- and $\mathbf{HB}^{=}$-derivations that was explained in the proof of Lemma 5.1.19.

■

The next theorem asserts an analogous correspondence between $\mathbf{AK}_{\mathbf{0}}^{=}$-consistency and recursive type equality as the one stated by Theorem 5.2.6 between $\mathbf{AK}^{=}$-consistency and $=_\mu$. For its formulation, we need to define the binary relation $\mathrm{Cons}_{\mathbf{AK}_{\mathbf{0}}^{=}}(\cdot,\cdot)$ on μTp: we do so by stipulating, for all $\tau, \sigma \in \mu Tp$,

$$\mathrm{Cons}_{\mathbf{AK}_{\mathbf{0}}^{=}}(\tau, \sigma) \quad \Longleftrightarrow_{\text{def}} \quad \tau = \sigma \text{ is } \mathbf{AK}_{\mathbf{0}}^{=}\text{-consistent}\ . \tag{5.17}$$

Theorem 5.2.13 (Correspondence between $\mathbf{AK}_{\mathbf{0}}^{=}$-consistency and $=_\mu$). *The relation $Cons_{\mathbf{AK}_{\mathbf{0}}^{=}}(\cdot,\cdot)$ on μTp coincides with recursive type equality $=_\mu$, i.e. for all $\tau, \sigma \in \mu Tp$ it holds:*

$$\tau = \sigma \textit{ is } \mathbf{AK}_{\mathbf{0}}^{=}\textit{-consistent} \iff \tau =_\mu \sigma\ . \tag{5.18}$$

Sketch of Proof. It suffices to show (5.18) for all $\tau, \sigma \in \mu Tp$. Both directions in this equation can be shown similarly as sketched above for a proof of Theorem 5.2.6, i.e. both the statements (5.12) and (5.13) from this proof hold with respect to the system $\mathbf{AK}_{\mathbf{0}}^{=}$ as well.

However, for the direction "$\Leftarrow$" the following stronger and more specific statement can be used which expresses the noteworthy fact that every derivation $\mathcal{D}$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ from assumption $\tau = \sigma$ corresponds to an access path p in both $\mathbf{Tree}(\tau)$ and $\mathbf{Tree}(\sigma)$ with the property that $\mathbf{Tree}(\tau)$ and $\mathbf{Tree}(\sigma)$ do not differ along any

[14] Since in the graphical representation of $\tilde{\mathcal{D}}$ as typeset in (5.16) it has actually not been specified in which order the first two applications of FOLD_l and FOLD_r do actually take place, the derivation $\mathcal{D}$ is in fact the translation to $\mathbf{AK}^{=}$ of that one of the two derivations in $\mathbf{AK}_{\mathbf{0}}^{=}$ denoted by the symbolic prooftree (5.16) in which the rule used in the topmost rule application is UNFOLD_l.

shorter access path $p_0 < p$:

$$\left.\begin{array}{l}
\big(\forall\,\tau,\sigma,\chi_1,\chi_2 \in \mu Tp\big)\\
\quad\left(\forall\mathcal{D}\ \begin{array}{l}\mathbf{AK}_{\mathbf{0}}^{=}\text{-derivation with (single) assumption}\\ \tau=\sigma\text{, with conclusion } \chi_1=\chi_2 \text{ and of the form}\end{array}\ \begin{array}{c}(\tau=\sigma)\\ \mathcal{D}\\ \chi_1=\chi_2\end{array}\right)\\
\qquad\big(\exists\, p\in Acc(\tau)\cap Acc(\sigma)\big)\\
\qquad\quad\Big[\,(\forall p_0\in\{1,2\}^*)\,\big[\,p_0<p \;\Rightarrow\; \mathbf{Tree}(\tau)(p_0)=\mathbf{Tree}(\sigma)(p_0)\,\big]\\
\qquad\qquad \&\ \ \big[\mathbf{Tree}(\tau)|_p=\mathbf{Tree}(\chi_1)\ \ \&\ \ \mathbf{Tree}(\sigma)|_p=\mathbf{Tree}(\chi_2)\big]\,\Big].
\end{array}\right\}\tag{5.19}$$

□

Remark 5.2.14. In Chapter 6, Section 6.3, we will encounter a similar correspondence result between a notion related to $\mathbf{AK}_{\mathbf{0}}^{=}$-consistency and recursive type equality: there, Corollary 6.3.19 asserts that the existence of a "consistency-unfolding in $\mathbf{AK}_{\mathbf{0}}^{=}$" (a notion defined in Section 6.3 of Chapter 6) for an equation $\tau = \sigma$ is equivalent to the recursive types τ and σ being strongly equivalent. In contrast with Theorem 5.2.13, for which the comments in item (b) of Remark 5.2.7 apply, Corollary 6.3.19 will be very much in line with most soundness and completeness theorems in logic. This is due to the fact that the notion "existence of a consistency-unfolding in $\mathbf{AK}_{\mathbf{0}}^{=}$" will be defined very similarly to the notion of provability in a natural-deduction-style proof system; and in particular, it is positively calculable because it allows for effective positive tests to take place.

As mentioned earlier, it is not possible to 'normalize' arbitrary derivations in $\mathbf{AK}^{=}$ into mimicking derivations in $\mathbf{AK}_{\mathbf{0}}^{=}$ because of the fact that, contrasting with $\mathbf{AK}_{\mathbf{0}}^{=}$-derivations, $\mathbf{AK}^{=}$-derivations may contain more assumptions than just one. But a transformation in the opposite direction is in fact possible, analogously as there is an easy transformation from $\mathbf{HB}^{=}$-derivations into mimicking $\mathbf{HB}_{\mathbf{0}}^{=}$-derivations as stated by Lemma 5.1.19 and demonstrated in its proof.

Lemma 5.2.15. *All rules of* $\mathbf{AK}_{\mathbf{0}}^{=}$ *are derivable in* $\mathbf{AK}^{=}$. *Every derivation* $\mathcal{D}$ *in* $\mathbf{AK}_{\mathbf{0}}^{=}$ *with (single) assumption* $\chi_1 = \chi_2$, *for some* $\chi_1, \chi_2 \in \mu Tp$, *can be transformed effectively into a derivation* $\mathcal{D}'$ *in* $\mathbf{AK}^{=}$ *that has the same conclusion and that contains, equally as* $\mathcal{D}$, *a single occurrence of* $\chi_1 = \chi_2$ *as an assumption.*

Sketch of Proof. Analogously to item (a) in the proof of Lemma 5.1.19, it is easy to show that all rules of $\mathbf{AK}_{\mathbf{0}}^{=}$ are derivable in $\mathbf{AK}^{=}$ (for instance, an arbitrary application ι of the rule FOLD_r can be mimicked by a derivation that consists of an application of TRANS below an axiom (FOLD/UNFOLD) in the left premise and the formula $\mathsf{prem}(\iota)$ in the left premise). And as a consequence, all applications of the rules $(\mu\!-\!\bot)_{l/r}^{\mathrm{der}\bot}$, $\mathrm{UNFOLD}_{l/r}$, and REN can be eliminated successively from an arbitrary derivation in $\mathbf{AK}_{\mathbf{0}}^{=}$ with the result of a mimicking derivation in $\mathbf{AK}^{=}$.

□

The following proposition formulates the obvious consequence of Lemma 5.2.15 that the consistency with respect to $\mathbf{AK}^{=}$ of an equation between recursive types implies the consistency of this equation with respect to $\mathbf{AK}_{\mathbf{0}}^{=}$.

Proposition 5.2.16. *Let $\tau, \sigma \in \mu Tp$. If the formula $\tau = \sigma$ is consistent with respect to $\mathbf{AK}^{=}$, then it is also consistent with respect to $\mathbf{AK}_{\mathbf{0}}^{=}$.*

Concluding this section, we give the following easy corollary that formulates, for the system $\mathbf{AK}_{\mathbf{0}}^{=}$, a statement analogous to the generalized version Theorem 5.2.8 of Theorem 5.2.6.

Corollary 5.2.17. *Let $\Gamma \subseteq \mu Tp\text{–}Eq$ be an arbitrary set of equations between recursive types. Then the following holds:*

$$\Gamma \textit{ is consistent with } \mathbf{AK}_{\mathbf{0}}^{=} \quad \Longleftrightarrow \quad (\forall\, \tau = \sigma \,\in\, \Gamma)\, \big[\, \tau =_{\mu} \sigma \,\big]\, . \tag{5.20}$$

Proof. As an immediate consequence of the fact that $\mathbf{AK}_{\mathbf{0}}^{=}$-derivations can only contain one assumption (since rules in $\mathbf{AK}_{\mathbf{0}}^{=}$ do not allow more-premise applications), a set Γ of formulas of μTp is $\mathbf{AK}_{\mathbf{0}}^{=}$-consistent if and only if every member of Γ is $\mathbf{AK}_{\mathbf{0}}^{=}$-consistent. With this the assertion of the lemma follows from the equivalences (5.18), for all $\tau, \sigma \in \mu Tp$, in Theorem 5.2.13. □

5.3 Basic Differences between the Axiom Systems and the Systems for Consistency-Checking

In this section we want to provide some facts that underline the substantial difference between the axiom systems for $=_{\mu}$ introduced in Section 5.1 and the systems for consistency-checking with respect to $=_{\mu}$ described in Section 5.2. We will give some basic observations about the different behaviour of these two kinds of systems with respect to the notions of *consistency* relative to them and of *formula derivability* in them. We will also investigate what the relationships are, on the one hand, of the distinctive *decomposition rule* DECOMP from the syntactic-matching systems to the axiom systems, and on the other hand, of the *composition rule* ARROW from the axiom systems to the syntactic-matching systems.

Our first goal is to clear up what notions of consistency are defined by the axiom systems from Section 5.1. In Definition 5.2.3, we have defined the notions of consistency and inconsistency of single formulas and of sets of formulas in μTp relative to a given proof system with formulas $\mu Tp\text{–}Eq$. In Theorem 5.2.6 we saw that the recursive types $\tau, \sigma \in \mu Tp$, for which the equation $\tau = \sigma$ is consistent with the syntactic-matching system $\mathbf{AK}^{=}$, are precisely those that are strongly equivalent. By Theorem 5.2.13 the same is true for the system $\mathbf{AK}_{\mathbf{0}}^{=}$. And Theorem 5.2.8 stated, that the sets of formulas $\Gamma \subseteq \mu Tp\text{–}Eq$, which are consistent with either of $\mathbf{AK}^{=}$ or $\mathbf{AK}_{\mathbf{0}}^{=}$, are just the sets consisting exclusively of equations between strongly equivalent recursive types. In connection with this we want to raise the question what kind of formulas and of sets of formulas in $\mu Tp\text{–}Eq$ are actually consistent with each of the axiom systems $\mathbf{AC}^{=}$, $\mathbf{AC}_{*}^{=}$, $\mathbf{HB}^{=}$, and $\mathbf{HB}_{\mathbf{0}}^{=}$.

For settling this question, the following lemma will turn out to be helpful.

Lemma 5.3.1. *Let $\mathcal{S}$ be one of the systems* $\mathbf{AC}^{=}$, $\mathbf{AC}^{=}_{*}$, $\mathbf{HB}^{=}$, *or* $\mathbf{HB}^{=}_{\mathbf{0}}$.

(i) Let $\tau = \sigma$ be an arbitrary axiom in $\mathcal{S}$. Then $\mathcal{L}'(\tau) = \mathcal{L}'(\sigma)$ holds, i.e. τ and σ have the same leading symbol.

(ii) Let $\dfrac{\overset{\mathcal{D}_1}{\tau_1 = \sigma_1} \qquad \overset{\mathcal{D}_2}{(\,\tau_2 = \sigma_2\,)}}{\tau = \sigma}\,R$ *be a derivation in $\mathcal{S}$, where $\mathcal{D}_1, \mathcal{D}_2 \in$ $\in \mathcal{D}er(\mathcal{S})$ and $\tau, \sigma, \tau_1, \tau_2, \sigma_1, \sigma_2 \in \mu Tp$, that ends with a one-(or two-)premise application ι of a rule R of $\mathcal{S}$. Then the leading symbols of the recursive types in the conclusion of ι are equal, given that the leading symbols of the recursive types in the equation(s) of the premise(s) of ι are equal, i.e. it holds:*

$$\mathcal{L}'(\tau_1) = \mathcal{L}'(\sigma_1) \;\; (\,\&\;\; \mathcal{L}'(\tau_2) = \mathcal{L}'(\sigma_2)\,) \quad\Longrightarrow\quad \mathcal{L}'(\tau) = \mathcal{L}'(\sigma)\ .$$

(iii) Let $\tau, \sigma \in \mu Tp$, $n \in \omega$, and $\tau_i, \sigma_i \in \mu Tp$ for $i \in \{1, \ldots, n\}$. Let furthermore $\overset{\mathcal{D}}{\tau = \sigma}$ *be a derivation in $\mathcal{S}$ with conclusion $\tau = \sigma$ and with assumptions according to:*

- *If $\mathcal{S}$ is* $\mathbf{AC}^{=}$ *of* $\mathbf{AC}^{=}_{*}$ *then set*$(\mathsf{assm}(\mathcal{D})) = \{\tau_1 = \sigma_1, \ldots, \tau_n = \sigma_n\}$, *and*
- *if $\mathcal{S}$ is* $\mathbf{HB}^{=}$ *of* $\mathbf{HB}^{=}_{\mathbf{0}}$ *then* $\mathsf{omassm}(\mathcal{D}) = \{(\tau_1 = \sigma_1)^{u_1}, \ldots, (\tau_n = \sigma_n)^{u_n}\}$ *for some $u_1, \ldots, u_n \in Mk$.*

Then the leading symbols in the conclusion of $\mathcal{D}$ are equal under the assumption that, for all assumptions $\tau_i = \sigma_i$ of $\mathcal{D}$, the leading symbols of τ_i and σ_i are equal, i.e. it holds:

$$(\,\forall\, i \in \{1, \ldots, n\}\,)\, [\,\mathcal{L}'(\tau_i) = \mathcal{L}'(\sigma_i)\,] \quad\Longrightarrow\quad \mathcal{L}'(\tau) = \mathcal{L}'(\sigma)\ .$$

Sketch of Proof. Verifying items (i) and (ii) of the lemma is mostly trivial and always easy with respect to all axioms and rules in one of the systems $\mathbf{AC}^{=}$, $\mathbf{AC}^{=}_{*}$, $\mathbf{HB}^{=}$, and $\mathbf{HB}^{=}_{\mathbf{0}}$ except for the treatment of the rules μ-COMPAT and CONTRACT. For these two rules the proof of item (ii) can be settled by case-distinction arguments on the leading symbol of the recursive types in the premise of an application.

The assertion in item (iii) follows from the items (i) and (ii) by straightforward induction on the depth $|\mathcal{D}|$ of $\mathcal{S}$-derivations. □

In the following theorem a characterization will be given of the notions of consistency relative to each of the systems $\mathbf{AC}^{=}$, $\mathbf{AC}^{=}_{*}$, $\mathbf{HB}^{=}$, and $\mathbf{HB}^{=}_{\mathbf{0}}$. It turns out that all of these notions of consistency do actually coincide and that they differ from the, also coinciding, notions of consistency with respect to the syntactic-matching systems $\mathbf{AK}^{=}$ and $\mathbf{AK}^{=}_{\mathbf{0}}$.

Theorem 5.3.2 (Characterization of the consistent formulas relative to $\mathbf{AC}^{=}$, $\mathbf{AC}^{=}_{*}$, $\mathbf{HB}^{=}$, and $\mathbf{HB}^{=}_{\mathbf{0}}$). *Let $\mathcal{S}$ be either of the systems $\mathbf{AC}^{=}$, $\mathbf{AC}^{=}_{*}$, $\mathbf{HB}^{=}$, or $\mathbf{HB}^{=}_{\mathbf{0}}$. Then, for all $\tau, \sigma \in \mu Tp$ and for all sets $\Gamma \subseteq \mu Tp\text{–}Eq$, the following assertions hold:*

$$\tau = \sigma \ \textit{ is } \mathcal{S}\textit{-consistent} \quad \Longleftrightarrow \quad \mathcal{L}'(\tau) = \mathcal{L}'(\sigma) \ ; \tag{5.21}$$

$$\Gamma \ \textit{ is } \mathcal{S}\textit{-consistent} \quad \Longleftrightarrow \quad (\forall\, \tilde{\tau} = \tilde{\sigma} \ \in \Gamma\,)\, [\, \mathcal{L}'(\tilde{\tau}) = \mathcal{L}'(\tilde{\sigma})\,]\,. \tag{5.22}$$

Proof. The directions "⇒" in (5.21) and in (5.22) are trivial; the directions "⇐" follow directly from Lemma 5.3.1, (iii). □

Thus we have found that the formulas from $\mu Tp\text{–}Eq$ that are consistent with an arbitrary one of the axiom systems in Section 5.1 are precisely those equations between recursive types that are not themselves contradictions w.r.t. $=_{\mu}$. Hence, for each of the axiom systems $\mathcal{S}$ described here, the notions of "consistency relative to $\mathcal{S}$" for single formulas and for sets of formulas are essentially *trivial* and *much weaker* than the notion of consistency relative to the syntactic-matching systems $\mathbf{AK}^{=}$ or $\mathbf{AK}^{=}_{\mathbf{0}}$ (to which systems, as we have seen before, only equations between recursive types $\tau = \sigma$ for τ and σ that have the same tree unfolding $\mathbf{Tree}(\tau) = \mathbf{Tree}(\sigma)$ can be added consistently).

We continue by studying properties of the decomposition rule DECOMP from the syntactic-matching systems in relation to the axiom systems $\mathbf{AC}^{=}$ and $\mathbf{AC}^{=}_{*}$.

Theorem 5.3.3 (Relationship of DECOMP to the systems $\mathbf{AC}^{=}$ and $\mathbf{AC}^{=}_{*}$). *The rule* DECOMP *of $\mathbf{AK}^{=}$ and $\mathbf{AK}^{=}_{\mathbf{0}}$ is admissible, but not derivable in the systems $\mathbf{AC}^{=}$ and $\mathbf{AC}^{=}_{*}$.*

Proof. Let $\mathcal{H}$ be one of the systems $\mathbf{AC}^{=}$ or $\mathbf{AC}^{=}_{*}$. We will show that DECOMP is admissible in $\mathcal{H}$ and that it is not derivable in $\mathcal{H}$ in the below items (a) and (b), respectively.

(a) We are going to prove that DECOMP is an admissible rule in $\mathcal{H}$ by showing that DECOMP is correct for $\mathcal{H}$ (this is sufficient because of Proposition 4.2.4, (i)). For this, we will use the soundness and completeness theorems of $\mathcal{H}$ with respect to $=_{\mu}$, Theorem 5.1.4 and Theorem 5.1.9.

Let $\dfrac{\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2}{\tau_i = \sigma_i}\,\textsc{decomp}$, where $\tau_1, \tau_2, \sigma_1, \sigma_2 \in \mu Tp$, be an arbitrary application of DECOMP such that $\vdash_{\mathcal{H}} \tau_1 \to \tau_2 = \sigma_1 \to \sigma_2$ holds. We have to show that also $\vdash_{\mathcal{H}} \tau_i = \sigma_i$ holds. Since $\mathcal{H}$ is sound with respect to $=_{\mu}$, we find that $\tau_1 \to \tau_2 =_{\mu} \sigma_1 \to \sigma_2$ and hence also that $\mathbf{Tree}(\tau_1 \to \tau_2) = = \mathbf{Tree}(\sigma_1 \to \sigma_2)$ holds. By the definition of the tree unfolding it follows that $\mathbf{Tree}(\tau_i) = \mathbf{Tree}(\sigma_i)$ and therefore also that $\tau_i =_{\mu} \sigma_i$ holds. And from this we obtain $\vdash_{\mathcal{H}} \tau_i = \sigma_i$ by completeness of $\mathcal{H}$ with respect to $=_{\mu}$.

(b) We are going to prove that DECOMP is not a derivable rule in $\mathcal{H}$.

For this, we *assume* that DECOMP is a derivable rule of $\mathcal{H}$ and will infer a contradiction with Lemma 5.3.1. We choose $\alpha, \beta \in \mathit{TVar}$ arbitrarily, and consider an application ι of DECOMP of the form $\dfrac{\alpha \to \bot = \alpha \to \beta}{\bot = \beta}\,\textsc{decomp}$.

By our assumption there exists a derivation of $\mathcal{H}$ that mimics ι. We choose one such derivation $\mathcal{D}^{(\iota)}_{\mathrm{mim}} \in \mathcal{D}er(\mathcal{H})$. Since $\mathcal{D}^{(\iota)}_{\mathrm{mim}}$ mimics ι, it must be of the form

$$\begin{array}{c}[\alpha \to \bot = \alpha \to \beta]\\ \mathcal{D}^{(\iota)}_{\mathrm{mim}}\\ \bot = \beta\end{array}$$

with assumptions $\mathrm{set}(\mathsf{assm}(\mathcal{D}^{(\iota)}_{\mathrm{mim}})) = \{\alpha \to \bot = \alpha \to \beta\}$ because it cannot be the case that $\mathrm{set}(\mathsf{assm}(\mathcal{D}^{(\iota)}_{\mathrm{mim}})) = \emptyset$ holds: otherwise $\vdash_{\mathcal{H}} \bot = \beta$ would follow in contradiction with the fact that actually $\not\vdash_{\mathcal{H}} \bot = \beta$ holds due to $\bot \neq_{\mu} \beta$ and the soundness of $\mathcal{H}$ with respect to $=_{\mu}$. But the existence of such a derivation $\mathcal{D}^{(\iota)}_{\mathrm{mim}}$ contradicts item (iii) of Lemma 5.3.1 due to the facts $\mathcal{L}'(\alpha \to \bot) = \to = \mathcal{L}'(\alpha \to \beta)$ and $\mathcal{L}'(\bot) = \bot \neq \beta = \mathcal{L}'(\beta)$. Hence our assumption, that DECOMP is a derivable rule of $\mathcal{H}$, cannot be upheld.

□

The question about the relationship of the rule DECOMP in $\mathbf{AK}^{=}$ to each of the Brandt-Henglein systems $\mathbf{HB}^{=}$ and $\mathbf{HB}^{=}_{\mathbf{0}}$ cannot be answered analogously, due to the fact that DECOMP is a pure-Hilbert-system rule whereas $\mathbf{HB}^{=}$ and $\mathbf{HB}^{=}_{\mathbf{0}}$ are natural-deduction systems. However, DECOMP can be adapted, albeit in different ways, for use in natural-deduction systems. Among its reformulations as natural-deduction-system rules there are the following two obvious possibilities: the rule $\mathrm{DECOMP}^{-}_{(\mathrm{nd})}$ with applications of the form

$$\frac{\begin{array}{c}\mathcal{D}_1\\ \tau_1 \to \tau_2 = \sigma_1 \to \sigma_2\end{array}}{\tau_i = \sigma_i}\,\mathrm{DECOMP}^{-}_{(\mathrm{nd})} \quad (\text{no side-condition on } \mathcal{D}_1)\ , \tag{5.23}$$

in which the presence, or absence, of assumptions in immediate subderivations is 'ignored'; and the rule $\mathrm{DECOMP}_{(\mathrm{nd})}$ with applications of the form

$$\frac{\begin{array}{c}\mathcal{D}_1\\ \tau_1 \to \tau_2 = \sigma_1 \to \sigma_2\end{array}}{\tau_i = \sigma_i}\,\mathrm{DECOMP}_{(\mathrm{nd})} \quad (\text{if } \mathsf{omassm}(\mathcal{D}_1) = \emptyset)\ , \tag{5.24}$$

which rule can only be applied to a derivation $\mathcal{D}_1$ in the absence of open assumptions in $\mathcal{D}_1$. Clearly, every application of the rule $\mathrm{DECOMP}_{(\mathrm{nd})}$ can be mimicked by a corresponding application of the rule $\mathrm{DECOMP}^{-}_{(\mathrm{nd})}$, but the opposite is not always the case. This has the consequence that admissibility, cr-admissibility, or derivability of $\text{DECOMP}^{-}_{(\text{nd})}$ in a natural-deduction system $\mathcal{S}$ respectively implies admissibility, cr-admissibility, or derivability of $\text{DECOMP}_{(\text{nd})}$ in $\mathcal{S}$, but also that the converse implications do not need to be true for all such systems. The theorem below is the outcome of an examination of the relationships of these two natural-deduction system variants to the systems $\mathbf{HB}^{=}$ and $\mathbf{HB}^{=}_{\mathbf{0}}$ with respect to the rule derivability and (cr-)admissibility. Incidentally, the theorem also states that $\text{DECOMP}^{-}_{(\text{nd})}$ and $\text{DECOMP}_{(\text{nd})}$ do in fact differ with respect to admissibility and cr-admissibility in the systems $\mathbf{HB}^{=}$ and $\mathbf{HB}^{=}_{\mathbf{0}}$.

Theorem 5.3.4 (Relationship of DECOMP to the systems $\mathbf{HB}^{=}$ and $\mathbf{HB}^{=}_{\mathbf{0}}$). *For the natural-deduction system variants* $\text{DECOMP}^{-}_{(\text{nd})}$ *and* $\text{DECOMP}_{(\text{nd})}$ *of the rule* DECOMP *in* $\mathbf{AK}^{=}$ *the following statements hold:*

(i) The rule $\text{DECOMP}^{-}_{(\text{nd})}$ *is not admissible, let alone cr-admissible or derivable, in* $\mathbf{HB}^{=}$ *and in* $\mathbf{HB}^{=}_{\mathbf{0}}$*.*

(ii) The rule $\text{DECOMP}_{(\text{nd})}$ *is cr-admissible, and hence admissible, but not derivable in* $\mathbf{HB}^{=}$ *and in* $\mathbf{HB}^{=}_{\mathbf{0}}$*.*

Proof. We will prove assertions (i) and (ii) of the theorem in items (a) and (b) below. For doing this, we let $\mathcal{S}$ be an arbitrary one of the systems $\mathbf{HB}^{=}$ or $\mathbf{HB}^{=}_{\mathbf{0}}$.

(a) Since due to Lemma 4.3.5, (ii), rule derivability and rule cr-admissibility are stronger than rule admissibility, we only have to show that $\text{DECOMP}^{-}_{(\text{nd})}$ is not admissible in $\mathcal{S}$. For this we will show

$$\mathcal{S} + \text{DECOMP}^{-}_{(\text{nd})} \not\sim_{th} \mathcal{S} \,, \tag{5.25}$$

i.e. that there exist theorems of $\mathcal{S}$+$\text{DECOMP}^{-}_{(\text{nd})}$ which are not theorems of $\mathcal{S}$. As a first observation, we notice that, for all $\tau_1, \tau_2, \sigma_1, \sigma_2 \in \mu Tp$, the derivation

$$\dfrac{\dfrac{(\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2)^u}{\tau_1 = \sigma_1}\,\text{DECOMP}^{-}_{(\text{nd})} \qquad \dfrac{(\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2)^u}{\tau_2 = \sigma_2}\,\text{DECOMP}^{-}_{(\text{nd})}}{\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2}\,\text{ARROW/FIX}, u$$

in $\mathcal{S}$+$\text{DECOMP}^{-}_{(\text{nd})}$ does not contain open assumption classes. Hence all formulas consisting of equations between composite types are theorems of $\mathcal{S}$+$\text{DECOMP}^{-}_{(\text{nd})}$. And in particular, we find that

$$\vdash_{\mathcal{S}+\text{DECOMP}^{-}_{(\text{nd})}} \alpha \to \bot = \alpha \to \beta$$

holds for all type variables α and β. However, we also find $\not\vdash_{\mathcal{S}} \alpha \to \bot = \alpha \to \beta$ for all $\alpha, \beta \in TVar$ as a consequence of $\alpha \to \bot \neq_{\mu} \alpha \to \beta$ and the soundness[15]

[15] The use of the soundness theorem for $\mathcal{S}$ could be circumvented here by showing directly that the formula $\alpha \to \bot = \alpha \to \beta$ is not a theorem of $\mathcal{S}$. In the case $\mathcal{S} = \mathbf{HB}^{=}_{\mathbf{0}}$, this can be shown in a particularly easy way, as follows. From every derivation $\mathcal{D}$ in $\mathbf{HB}^{=}_{\mathbf{0}}$ with $\mathsf{concl}(\mathcal{D}) = \alpha \to \bot = \alpha \to \beta$ and $\mathsf{omassm}(\mathcal{D}) = \emptyset$ a derivation $\mathcal{D}_a$ in $\mathbf{HB}^{=}_{\mathbf{0}}$ with $\mathsf{concl}(\mathcal{D}_a) = \bot \to \beta$ and $\mathsf{omassm}(\mathcal{D}_a) = \emptyset$ can be extracted in a straightforward way. But such a derivation $\mathcal{D}_a$ cannot exist in $\mathbf{HB}^{=}_{\mathbf{0}}$ due to Lemma 5.3.1.

of $=_\mu$ with respect to $\mathcal{S}$. Thus we have demonstrated (5.25), and hence $\text{DECOMP}^{-}_{(\text{nd})}$ is not admissible in $\mathcal{S}$.

(b) In view of Lemma 4.3.5 it suffices to show that $\text{DECOMP}_{(\text{nd})}$ is cr-correct for $\mathcal{S}$ (because this implies cr-admissibility and admissibility of $\text{DECOMP}_{(\text{nd})}$ in these systems) and that R is not derivable in $\mathcal{S}$. We demonstrate these statements subsequently below.

For showing cr-correctness of $\text{DECOMP}_{(\text{nd})}$ for $\mathcal{S}$, we will argue, similar as in item (a) of the proof of Theorem 5.3.3, by using the soundness and completeness theorems of $\mathcal{S}$ with respect to $=_\mu$. Let $\mathcal{D}$ be a derivation in $\mathcal{S}+\text{DECOMP}_{(\text{nd})}$ of the form (5.24), where $\tau_1, \tau_2, \sigma_1, \sigma_2 \in \mu Tp$, $i \in \{1,2\}$, and $\mathcal{D}_1$ is a derivation in $\mathcal{S}$ without open assumptions. We have to show the existence of a derivation $\mathcal{D}'$ in $\mathcal{S}$ without assumptions and with conclusion $\tau_i = \sigma_i$, i.e. we have to show $\vdash_{\mathcal{S}} \tau_i = \sigma_i$. Since $\mathsf{omassm}(\mathcal{D}_1) = \emptyset$ holds, we find $\vdash_{\mathcal{S}} \tau_1 \to \tau_2 = \sigma_1 \to \sigma_2$. Due to the soundness of $\mathcal{S}$ with respect to $=_\mu$, we find $\tau_1 \to \tau_2 =_\mu \sigma_1 \to \sigma_2$, which implies (due to the definition of $=_\mu$ via the tree unfolding) $\tau_i =_\mu \sigma_i$. By completeness of $\mathcal{S}$, $\vdash_{\mathcal{S}} \tau_i = \sigma_i$ follows.

Now we show that $\text{DECOMP}_{(\text{nd})}$ is not derivable in $\mathcal{S}$. For this, we assume that $\text{DECOMP}_{(\text{nd})}$ is a derivable rule in $\mathcal{S}$ and show that this leads to a contradiction.

We designate by $\mathcal{S}_{\text{ext}}$ be the extension by enlargement of $\mathcal{S}$ by adding the rule $\text{DECOMP}_{(\text{nd})}$ as well as the zero-premise rule with the single application

$$\frac{}{\alpha \to \beta = \alpha \to \gamma}\ \text{EQ}^{(\alpha\to\beta,\,\alpha\to\gamma)}$$

where α, β and γ are specific type variables in $TVar$; this rule is obviously unsound with respect to $=_\mu$. However, since $\text{DECOMP}_{(\text{nd})}$ is, according to our assumption, derivable in $\mathcal{S}$, it follows that for the derivation $\mathcal{D}$ in $\mathcal{S}_{\text{ext}}$ that is of the form

$$\frac{\dfrac{}{\left[\alpha \to \beta = \alpha \to \gamma\right]_1}\ \text{EQ}^{(\alpha\to\beta,\,\alpha\to\gamma)}}{\beta = \gamma}\ \text{DECOMP}_{(\text{nd})}$$

there must exist a unary derivation context $\mathcal{DC}'$ in $\mathcal{S}$ such that the derivation $\mathcal{D}'$ of the form

$$\begin{array}{c} \dfrac{}{\left[\alpha \to \beta = \alpha \to \gamma\right]_1}\ \text{EQ}^{(\alpha\to\beta,\,\alpha\to\gamma)} \\ \mathcal{DC}' \\ \beta = \gamma \end{array}$$

is a derivation in $\mathcal{S}_{\text{ext}}$ without open assumptions. We choose $\mathcal{DC}'$ and $\mathcal{D}'$ in this way.

We observe that $\mathcal{DC}'$ must contain occurrences of the hole $[\,]_1$: otherwise $\mathcal{DC}'$ would itself be a derivation in $\mathcal{S}$ without open assumptions, and hence the conclusion of $\mathcal{DC}'$, the formula $\beta = \gamma$, would be a theorem of $\mathcal{S}$, which would contradict the soundness of $\mathcal{S}$ (any of the systems $\mathbf{HB}^{=}$ and $\mathbf{HB}_{\mathbf{0}}^{=}$) with respect to $=_\mu$.

Now we let u be an arbitrary assumption marker that does not occur in $\mathcal{DC}'$. And we note that, as a consequence of the observation in the last paragraph, the prooftree of the form

$$\begin{array}{c} \big[(\alpha \to \beta = \alpha \to \gamma)^u\big]_1 \\ \mathcal{DC}' \\ \beta = \gamma \end{array}$$

denotes a derivation $\mathcal{D}''$ in $\mathcal{S}$ with conclusion $\beta = \gamma$ and with the assumption class $[\alpha \to \beta = \alpha \to \gamma]^u$ as its single class of open assumptions, i.e. with

$$\mathsf{concl}(\mathcal{D}'') = \ (\beta = \gamma) \quad \& \quad \mathsf{omassm}(\mathcal{D}'') = \{(\alpha \to \beta = \alpha \to \gamma)^u\}\ . \qquad (5.26)$$

But due to $\mathcal{L}'(\alpha \to \beta) = \to = \mathcal{L}'(\alpha \to \gamma)$ and $\mathcal{L}'(\beta) = \beta \neq \gamma = \mathcal{L}'(\gamma)$ the existence of a derivation $\mathcal{D}''$ in $\mathcal{S}$ with (5.26) is a contradiction to Lemma 5.3.1, (iii). Therefore the assumption we made, that DECOMP$_{(\text{nd})}$ is derivable in $\mathcal{S}$, cannot be sustained.

□

As a foretaste of proof-theoretical transformations to be described in later chapters, we want to give also the following alternative proof for cr-admissibility of the rule DECOMP$_{(\text{nd})}$ in the variant-Brandt-Henglein system $\mathbf{HB}_{\mathbf{0}}^{=}$. In this proof we make use of the, compared to derivations in $\mathbf{HB}^{=}$, easier structure of derivations in $\mathbf{HB}_{\mathbf{0}}^{=}$, and we sketch how applications of DECOMP$_{(\text{nd})}$ can be effectively eliminated from derivations in $\mathbf{HB}_{\mathbf{0}}^{=}$+DECOMP$_{(\text{nd})}$.

Alternative proof for cr-admissibility of DECOMP$_{(\text{nd})}$ *in* $\mathbf{HB}_{\mathbf{0}}^{=}$. In order to show that DECOMP$_{(\text{nd})}$ is admissible in $\mathbf{HB}_{\mathbf{0}}^{=}$, it suffices, due to Lemma 4.3.5, (i), to show cr-correctness of DECOMP$_{(\text{nd})}$ for $\mathbf{HB}_{\mathbf{0}}^{=}$.

For showing cr-correctness of DECOMP$_{(\text{nd})}$ for $\mathbf{HB}_{\mathbf{0}}^{=}$, let $\mathcal{D}$ be an arbitrary derivation in $\mathbf{HB}_{\mathbf{0}}^{=}$+DECOMP$_{(\text{nd})}$ of the form (5.24), where $\tau_1, \tau_2, \sigma_1, \sigma_2 \in \mu Tp$, $i \in \{1, 2\}$, and $\mathcal{D}_1$ is a derivation in $\mathbf{HB}_{\mathbf{0}}^{=}$ without open assumptions and with conclusion $\tau_i = \sigma_i$. We have to demonstrate the existence of a derivation $\mathcal{D}'$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ without open assumptions and with conclusion $\tau_i = \sigma_i$. We will do so by case-distinction on the form of $\mathcal{D}$ in the remaining part of this proof.

If $\mathcal{D}_1$ does not contain any applications of ARROW or ARROW/FIX, then it is easy to see that $\mathcal{D}_1$ can only consist of an axiom (REFL) of the form $\rho_1 \to \rho_2 = = \rho_1 \to \rho_2$ for some $\rho_1, \rho_2 \in \mu Tp$, that is followed by zero, one or more REN-applications. Hence $\mathcal{D}_1$ is of the form

$$\begin{array}{c} (\text{REFL}) \\ \hline \rho_1 \to \rho_2 = \rho_1 \to \rho_2 \\ \hline\hline \tau_1 \to \tau_2 = \sigma_1 \to \sigma_2 \end{array}\ \text{REN}$$

and then $\mathcal{D}$ can be replaced by

$$\begin{array}{c} (\text{REFL}) \\ \hline \rho_i = \rho_i \\ \hline \tau_i = \sigma_i \end{array}\ \text{REN}$$

which is a derivation in $\mathbf{HB}_{\mathbf{0}}^{=}$ without open assumptions and with the same conclusion as $\mathcal{D}$, and which can therefore be taken as the desired derivation $\mathcal{D}'$.

If $\mathcal{D}_1$ contains at least one application of ARROW or ARROW/FIX, then it must be of either of the forms

$$\begin{array}{cl} \begin{array}{cc} \mathcal{D}_{1a1} & \mathcal{D}_{1a2} \\ \tau_1' = \sigma_1' & \tau_2' = \sigma_2' \end{array} & \\ \cline{1-1} \tau_1' \to \tau_2' = \sigma_1' \to \sigma_2' & \text{ARROW} \\ \cline{1-1}\cline{1-1} \tau_1 \to \tau_2 = \sigma_1 \to \sigma_2 & \text{REN} \end{array} \qquad \begin{array}{cl} \begin{array}{cc} [\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2]^u & [\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2]^u \\ \mathcal{D}_{1a1} & \mathcal{D}_{1a2} \\ \tau_1' = \sigma_1' & \tau_2' = \sigma_2' \end{array} & \\ \cline{1-1} \tau_1' \to \tau_2' = \sigma_1' \to \sigma_2' & \text{ARR./FIX, } u \\ \cline{1-1}\cline{1-1} \tau_1 \to \tau_2 = \sigma_1 \to \sigma_2 & \text{REN} \end{array}$$

where in each case we denote by $\mathcal{D}_{1a}$ the subderivation of $\mathcal{D}_1$ that ends in the application of the ARROW or ARROW/FIX displayed in the respective symbolic prooftree. In the left case, i.e. if the bottommost application of a two-premise rule in $\mathcal{D}_1$ is an application of ARROW, then we can take

$$\begin{array}{cl} \mathcal{D}_{1ai} & \\ \tau_i' = \sigma_i' & \\ \cline{1-1} \tau_i = \sigma_i & \text{REN} \end{array}$$

as the desired derivation $\mathcal{D}'$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ without open assumptions and with the same conclusion as $\mathcal{D}$. In the right case, i.e. if the bottommost application of a two-premise rule in $\mathcal{D}$ is an application of ARROW/FIX, we can take the derivation

$$\begin{array}{cl} \mathcal{D}_{1a} & \\ [\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2] & \\ \mathcal{D}_{1ai}' & \\ \tau_i' = \sigma_i' & \\ \cline{1-1} \tau_i = \sigma_i & \text{REN} \end{array} \tag{5.27}$$

in $\mathbf{HB}_{\mathbf{0}}^{=}$ as the desired derivation $\mathcal{D}'$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ without assumptions and with the same conclusion as $\mathcal{D}$ (hereby $\mathcal{D}_{1ai}'$ arises from $\mathcal{D}_{1ai}$ by appropriately renaming the markers for certain discharged assumption classes in order to make the 'substitution' of $\mathcal{D}_{1a}$ for the marked open assumptions $(\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2)^u$ in $\mathcal{D}_{1ai}'$ possible without giving rise to unwanted bindings, or confusion about bindings, of open assumptions in $\mathcal{D}_{1a}$ by applications of ARROW/FIX in $\mathcal{D}_{1ai}'$). Note that, although the assumption class $[\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2]^u$ is inhabited in at least one of $\mathcal{D}_{1a1}$ and $\mathcal{D}_{1a2}$, it can happen here that it is uninhabited in $\mathcal{D}_{1ai}$. In this case the symbolic prooftree (5.27) denotes just the derivation $\mathcal{D}_{1ai}$ that is extended at the bottom by the application of REN displayed in (5.27).

□

Next we are going to investigate the question what the theorems of the syntactic-matching systems defined in the previous section look like. For the system $\mathbf{AK}_{\mathbf{0}}^{=}$ the answer is actually trivial: this system contains neither contain axioms nor rules with zero-premise applications and hence it does not possess any theorems. But the situation is different for the system $\mathbf{AK}^{=}$. Here the answer that we will give consists of a characterization of the derivable formulas $\tau = \sigma$ in $\mathbf{AK}^{=}$ in terms of a conversion between τ and σ with respect to a certain reduction relation on μTp. For this we need to define first a couple of 'natural' reduction relations on recursive types.

Definition 5.3.5 (The four reduction relations $\rightarrow_{\textbf{out-unf}}$, $\rightarrow_{\textbf{out-}(\mu\text{--}\bot)'}$, $\rightarrow_{\textbf{ren/out-unf}}$, and $\rightarrow_{\textbf{r/o-u}(\mu\bot)'}$ on the set μTp). The reduction relations $\rightarrow_{\text{out-unf}}$, $\rightarrow_{\text{out-}(\mu\text{--}\bot)'}$, $\rightarrow_{\text{ren/out-unf}}$, and $\rightarrow_{\text{r/o-u}(\mu\bot)'}$ on μTp are formally defined as the following relations on $\mu Tp \times \mu Tp$:

$$\begin{aligned}
\rightarrow_{\text{out-unf}} &=_{\text{def}} \{ \langle \mu\alpha.\, \tau,\ \tau[\mu\alpha.\, \tau/\alpha] \rangle \mid \alpha \in TVar,\ \tau \in \mu Tp \} \\
\rightarrow_{\text{out-}(\mu\text{--}\bot)'} &=_{\text{def}} \{ \langle \mu\alpha_1 \ldots \alpha_n.\, \alpha_1,\ \bot) \rangle \mid n \in \omega\backslash\{0\}, \alpha_1 \ldots \alpha_n \in TVar \} \\
\rightarrow_{\text{ren/out-unf}} &=_{\text{def}} \ \rightarrow_{\text{out-unf}} \cup \rightarrow_{\text{ren}} \\
\rightarrow_{\text{r/o-u}(\mu\bot)'} &=_{\text{def}} \ \rightarrow_{\text{ren}} \cup \rightarrow_{\text{out-unf}} \cup \rightarrow_{\text{out-}(\mu\text{--}\bot)'}
\end{aligned}$$

(Clearly $\rightarrow_{\text{out-unf}}$ is the restriction of the unfolding-reduction $\rightarrow_{\text{unfold}}$ from Definition 3.7.7 to such unfoldings that take place at the *outermost* position in recursive types. The relation $\rightarrow_{\text{out-}(\mu\text{--}\bot)'}$ corresponds directly to the axioms $(\mu\!-\!\bot)'$ of $\mathbf{AK}^{=}$ and $\mathbf{HB}^{=}$; but it only allows the "use" of an axiom $(\mu\!-\!\bot)'$ at the *outermost* position[16] in a given recursive type $\tau \equiv \mu\alpha_1 \ldots, \alpha_n.\, \alpha_1$, when the whole term τ is reduced to $\bot$ in a reduction $\tau \rightarrow_{\text{out-unf}} \bot$).

According to the stipulations in Subsection 2.1.5, Chapter 2, we will denote more-step reduction relations and convertibility relations belonging to one of the reduction relations defined above with the use of double-headed arrows $\twoheadrightarrow$ and of the symbol $\twoheadleftarrow\!\!\!\twoheadrightarrow$: for instance $\twoheadleftarrow\!\!\!\twoheadrightarrow_{\text{r/o-u}(\mu\bot)'}$ will denote the *conversion* belonging to $\rightarrow_{\text{r/o-u}(\mu\bot)'}$, i.e. the reflexive, transitive and symmetrical closure of the relation $\rightarrow_{\text{r/o-u}(\mu\bot)'}$. ⊠

The following lemma concerns properties of the conversion $\twoheadleftarrow\!\!\!\twoheadrightarrow_{\text{r/o-u}(\mu\bot)'}$ that will be of use not only in this section, but also at some later occasions.

Lemma 5.3.6. *For all $\tau, \sigma, \tau_1, \tau_2, \sigma_1, \sigma_2 \in \mu Tp$ the assertions in the following three items are true:*

(i) $$\tau \twoheadleftarrow\!\!\!\twoheadrightarrow_{r/o\text{-}u(\mu\bot)'} \sigma \implies \tau \twoheadleftarrow\!\!\!\twoheadrightarrow_{ren/out\text{-}unf}\, \sigma \ \vee$$
$$\vee \ \ \tau, \sigma \in \left\{ \mu\alpha_1 \ldots \alpha_n.\, \begin{Bmatrix} \alpha_i \\ \bot \end{Bmatrix} \;\middle|\; \begin{array}{l} n \in \omega\backslash\{0\},\ 1 \le i \le n, \\ \alpha_1, \ldots, \alpha_n \in TVar \end{array} \right\}.$$

(Hereby $\begin{Bmatrix} \alpha_i \\ \bot \end{Bmatrix}$ in the set-expression of the right-hand side of the implication is intended to stand either for the term α_i or for the term $\bot$).

(ii) $$\tau \twoheadleftarrow\!\!\!\twoheadrightarrow_{ren/out\text{-}unf}\, \sigma \ \ \& \ \ nl\mu b(\tau) = nl\mu b(\sigma) = 0 \implies \tau \equiv_{ren} \sigma\ .$$

[16] This restriction is not in place for the similarly defined reduction $\rightarrow_{(\mu\text{--}\bot)}$ from Definition 3.7.7.

(iii) $\tau \;\twoheadleftarrow\!\!\twoheadrightarrow_{r/o\text{-}u(\mu\perp)'}\; \sigma \quad \& \quad nl\mu b(\tau) = nl\mu b(\sigma) = 0 \quad \Longrightarrow \quad \tau \equiv_{ren} \sigma \;.$

(iv) $\tau_1 \rightarrow \tau_2 \;\twoheadleftarrow\!\!\twoheadrightarrow_{r/o\text{-}u(\mu\perp)'}\; \sigma_1 \rightarrow \sigma_2 \quad \Longrightarrow \quad \tau_1 \equiv_{ren} \sigma_1 \;\;\&\;\; \tau_2 \equiv_{ren} \sigma_2 \;.$

Proof. (a) The assertion in item (i) of the lemma is a consequence of the following and a second related statement: for all $\tau \in \mu Tp$, $n \in \omega$ and $\beta_1, \ldots, \beta_n \in TVar$

$$\left.\begin{array}{l} \tau \;\rightarrow_{\text{out-unf}}\; \mu\beta_1 \ldots \beta_n.\, \beta_i \quad \Longrightarrow \\ \qquad\qquad \Longrightarrow \quad (\exists\, m \in \omega\backslash\{0\})\, (\exists\, \gamma_1, \ldots, \gamma_m \in TVar) \\ \qquad\qquad\qquad\qquad (\exists\, j \in \{1, \ldots, m\})\, \big[\, \tau \equiv \mu\gamma_1 \ldots \gamma_m.\, \gamma_j \,\big] \end{array}\right\} \tag{5.28}$$

holds; it is easy to prove this by a simple case-distinction. A second, related and even more obvious statement refers to reductions from a recursive type τ of the form $\tau \;\rightarrow_{\text{out-unf}}\; \mu\beta_1 \ldots \beta_n.\, \perp$, where $\beta_1, \ldots, \beta_n \in TVar$, and claims, that τ must then actually be of the form $\tau \equiv \mu\beta\beta_1 \ldots \beta_n.\, \perp$ for some $\beta \in TVar$. – From these two assertions it follows by an easy induction: whenever there is a conversion $\twoheadleftarrow\!\!\twoheadrightarrow_{\text{r/o-u}(\mu\perp)'}$ between recursive types τ and σ that contains a reduction $\rightarrow_{\text{out-}(\mu\text{–}\perp)'}$, then τ and σ must be contained in the set occurring in the disjunction on the right-hand side of the implication in item (i) of the lemma.

(b) For the proof of the assertion in item (ii), let $\tau, \sigma \in \mu Tp$ be arbitrary such that $\text{nl}\mu\text{b}(\tau) = \text{nl}\mu\text{b}(\sigma) = 0$ and $\tau \twoheadleftarrow\!\!\twoheadrightarrow_{\text{ren/out-unf}} \sigma$; we also choose one such conversion ξ between τ and σ.

If it were the case that there existed in ξ, for some $\alpha \in TVar$ and $\rho_0 \in \mu Tp$ such that $\alpha \not\downarrow \rho_0$, a reduction $\mu\alpha.\, \rho_0 \rightarrow_{\text{out-unf}} \rho_0[\mu\alpha.\, \rho_0/\alpha_0]$, then it followed with the help of (5.28) that every recursive type in ξ would be of the form $\mu\beta_1 \ldots \beta_n.\, \beta_i$ for some $n \in \omega\backslash\{0\}$, $i \in \{1, \ldots, n\}$ and variables $\beta_1, \ldots, \beta_n$; but this would contradict $\text{nl}\mu\text{b}(\tau) = \text{nl}\mu\text{b}(\sigma) = 0$. Hence for all unfolding-reductions $\rho_i \rightarrow_{\text{out-unf}} \rho_{i+1}$ that take place in ξ it holds (this follows from Lemma 3.5.7) that $\text{nl}\mu\text{b}(\rho_i) = \text{nl}\mu\text{b}(\rho_{i+1}) + 1$, and because of this the 'measure' $\text{nl}\mu\text{b}(\cdot)$ decreases strictly during each unfolding reduction-step in ξ.

It follows that each "local maximum" $m \in \omega$ with $m > 0$ of $\text{nl}\mu\text{b}(\cdot)$ within ξ occurs in a part of the form

$$\rho_0[\mu\alpha.\, \rho_0/\alpha] \leftarrow_{\text{out-unf}} \mu\alpha.\, \rho_0 \twoheadleftarrow\!\!\twoheadrightarrow_{\text{ren}} \mu\tilde{\alpha}.\, \tilde{\rho}_0 \rightarrow_{\text{out-unf}} \tilde{\rho}_0[\mu\tilde{\alpha}.\, \tilde{\rho}_0/\alpha] \tag{5.29}$$

(for some $\alpha, \tilde{\alpha} \in TVar$ and $\rho_0, \tilde{\rho}_0 \in \mu Tp$), where

$$\begin{aligned} m = \text{nl}\mu\text{b}(\mu\alpha.\, \rho_0) &= \text{nl}\mu\text{b}(\mu\tilde{\alpha}.\, \tilde{\rho}_0) = \\ &= \text{nl}\mu\text{b}(\rho_0[\mu\alpha.\, \rho_0/\alpha]) + 1 = \text{nl}\mu\text{b}(\tilde{\rho}_0[\mu\tilde{\alpha}.\, \tilde{\rho}_0/\tilde{\alpha}]) + 1 \;. \end{aligned}$$

It is now possible to eliminate successively each such local maximum from an already reached conversion between τ and σ by always replacing a part (5.29) by a part

$$\rho_0[\mu\alpha.\, \rho_0/\alpha] \twoheadrightarrow_{\text{ren}} \tilde{\rho}_0[\mu\tilde{\alpha}.\, \tilde{\rho}_0/\alpha] \;,$$

which replacements are justified by statement (3.19) of Lemma 3.4.2, in view of the fact that $\equiv_{\text{ren}}$ and $\twoheadrightarrow_{\text{ren}}$ coincide because $\rightarrow_{\text{ren}}$ is symmetric. In every elimination step of this kind the number of occurrences of $\rightarrow_{\text{out-unf}}$-reductions in a conversion between τ and σ decreases by 2. Since $\text{nl}\mu\text{b}(\tau) = \text{nl}\mu\text{b}(\sigma) = 0$, after finitely many elimination steps eventually a conversion $\twoheadleftarrow\!\!\twoheadrightarrow_{\text{ren/out-unf}}$ between τ and σ is reached, in which no occurring recursive type has a leading μ-binding. This conversion must then consist only of renaming-reductions $\rightarrow_{\text{ren}}$ and therefore we find $\tau \equiv_{\text{ren}} \sigma$.[17]

(c) The proof of (iii) is a direct consequence of the assertions in items (i) and (ii). And the assertion in (iv) follows from (iii) very obviously.

□

We are now finally in a position to offer an answer for the above posed question about the theorems of the syntactic-matching system $\mathbf{AK}^{=}$. Our answer has the form of the following characterization.

Theorem 5.3.7 (A characterization of the theorems of $\mathbf{AK}^{=}$). *The theorems of* $\mathbf{AK}^{=}$ *can be characterized as follows: For all* $\tau, \sigma \in \mu Tp$ *it holds that*

$$\vdash_{\mathbf{AK}^{=}} \tau = \sigma \quad \Longleftrightarrow \quad \tau \twoheadleftarrow\!\!\twoheadrightarrow_{r/o\text{-}u(\mu\bot)'} \sigma\ . \tag{5.30}$$

Proof. The implication "⇐" for all $\tau, \sigma \in \mu Tp$ follows by a straightforward induction, in which the assertion

$$(\forall\, \tau, \sigma \in \mu Tp) \left[\tau \rightarrow_{\text{r/o-u}(\mu\bot)'} \sigma \implies \vdash_{\mathbf{AK}^{=}} \tau = \sigma \right] \tag{5.31}$$

and the presence of the rules SYMM and TRANS in $\mathbf{AK}^{=}$ is used. (5.31) follows immediately from the presence of the axioms (REN), $(\mu\text{–}\bot)'$ and (FOLD/UNFOLD) in $\mathbf{AK}^{=}$.

The implication "⇒" for all $\tau, \sigma \in \mu Tp$ can be shown by induction on the depth $|\mathcal{D}|$ of an arbitrary given derivation $\mathcal{D}$ with conclusion $\tau = \sigma$ in $\mathbf{AK}^{=}$. Thereby the only non-trivial case to consider occurs in the induction step with $|\mathcal{D}| > 0$, when the last rule application in $\mathcal{D}$ is an application of DECOMP: in this case $\mathcal{D}$ is of the form

$$\dfrac{\begin{array}{c}\mathcal{D}_0\\ \tau_1 \rightarrow \tau_2 = \sigma_1 \rightarrow \sigma_2\end{array}}{\tau_i = \sigma_i}\ \text{DECOMP}$$

[17] It is perhaps interesting to note, that "local minima" with respect to $\text{nl}\mu\text{b}(\cdot)$ that occur in a situation analogously to (5.29) with $\rightarrow_{\text{out-unf}}$-reductions directed *to* the $\twoheadleftarrow\!\!\twoheadrightarrow_{\text{ren}}$-conversion in the middle cannot the eliminated from $\twoheadleftarrow\!\!\twoheadrightarrow_{\text{ren/out-unf}}$-conversions in a similar way. If this were the case, then a slight refinement of the above argument could be extended to a proof of a stronger statement than item (iii) with the weaker assumption $\text{nl}\mu\text{b}(\tau) = \text{nl}\mu\text{b}(\sigma)$ in its hypothesis instead of $\text{nl}\mu\text{b}(\tau) = \text{nl}\mu\text{b}(\sigma) = 0$. But such a stronger statement is actually false: for example with $\tau_1 \equiv \mu\alpha.\,(\alpha \rightarrow \bot)$ and $\chi \equiv \mu\beta.\,(\tau_1 \rightarrow \bot)$ for every $\beta \in TVar$ it holds that $\tau_1 \rightarrow_{\text{out-unf}} (\tau_1 \rightarrow \bot) \leftarrow_{\text{out-unf}} \chi$, (since $\beta \notin \text{fv}(\tau_1 \rightarrow \bot) = \emptyset$) and $\text{nl}\mu\text{b}(\tau_1) = \text{nl}\mu\text{b}(\chi) = 1$, but at the same time τ_1 and χ are no variants of each other.

with some $\tau_1, \tau_2, \sigma_1, \sigma_2 \in \mu Tp,\ i \in \{1,2\}$ and $\mathcal{D}_0$ an $\mathbf{AK^=}$-derivation. Here it follows from the induction hypothesis that $\tau_1 \to \tau_2 \twoheadleftrightarrow_{\mathrm{r/o\text{-}u}(\mu\bot)'} \sigma_1 \to \sigma_2$. But due to Lemma 5.3.6 (iv), we can conclude $\tau_i \equiv_{\mathrm{ren}} \sigma_i$ from this, which (since $\equiv_{\mathrm{ren}} = \ = \twoheadleftrightarrow_{\mathrm{ren}} \subseteq \twoheadleftrightarrow_{\mathrm{r/o\text{-}u}(\mu\bot)'}$) shows the desired conclusion $\tau_i \twoheadleftrightarrow_{\mathrm{r/o\text{-}u}(\mu\bot)'} \sigma_i$ of the induction step in this case.

□

Theorem 5.3.7 implies that the theorems of $\mathbf{AK^=}$ are only a very small and proper subset of all those equations $\tau = \sigma$ in $\mu Tp\text{–}Eq$ such that τ and σ are weakly equivalent. This is because $\twoheadleftrightarrow_{\mathrm{r/o\text{-}u}(\mu\bot)'} \subseteq \twoheadleftrightarrow_{(\mathrm{unf/ren}/\mu\bot)} = =_{w\mu}$ is the case (the set-inclusion "$\subseteq$" is justified by the fact, that $\mu\alpha_1 \ldots, \alpha_n.\, \alpha_1 \twoheadrightarrow_{(\mathrm{unf/ren}/\mu\bot)} \bot$ holds for all $n \in \omega \backslash \{0\}$ and $\alpha_1, \ldots, \alpha_n \in TVar$, and "$=$" is the assertion of Lemma 3.7.8), but $\twoheadleftrightarrow_{(\mathrm{unf/ren}/\mu\bot)} \neq =_{w\mu}$ is the case: $=_{w\mu}$ is compatible with μTp-contexts, whereas $\twoheadleftrightarrow_{\mathrm{r/o\text{-}u}(\mu\bot)'}$ is clearly not. Therefore it follows that $\twoheadleftrightarrow_{\mathrm{r/o\text{-}u}(\mu\bot)'} \subsetneqq =_{w\mu}$. Hence (formula) derivability in the syntactic-matching system $\mathbf{AK^=}$ is a *very much weaker notion* than even weak recursive type equality $=_{w\mu}$.

We conclude this section with a statement about the proof-theoretic relationship of the rule ARROW, contained in the systems $\mathbf{AC^=}$, $\mathbf{AC^=_*}$ and $\mathbf{HB^=_0}$ from Section 5.1, with respect to the syntactic-matching system $\mathbf{AK^=}$.

Corollary 5.3.8 (Relationship of the rule ARROW to the system $\mathbf{AK^=}$). *The rule* ARROW *is not admissible, let alone derivable, in* $\mathbf{AK^=}$*.*

Proof. Since due to Proposition 4.2.4, rule correctness as well as rule admissibility coincide and rule derivability is stronger than rule admissibility *in pure Hilbert systems*, we only have to demonstrate that ARROW is not correct for the pure Hilbert system $\mathbf{AK^=}$.

For this we *assume* that ARROW is correct for $\mathbf{AK^=}$ and will infer a contradiction with the characterization (5.30) of the theorems of $\mathbf{AK^=}$. As a consequence of our assumption, for the derivation $\mathcal{D}$

$$\dfrac{\dfrac{(\mu\text{–}\bot)}{\mu\alpha.\,\alpha = \bot} \qquad \dfrac{(\mathrm{REFL})}{\gamma = \gamma}}{(\mu\alpha.\,\alpha) \to \gamma = \bot \to \gamma}\ \mathrm{ARROW}$$

in $\mathbf{AK^=} + \mathrm{ARROW}$ there exists a derivation $\mathcal{D}'$ in $\mathbf{AK^=}$ with the same conclusion as $\mathcal{D}$ and without assumptions. Hence

$$\vdash_{\mathbf{AK^=}} (\mu\alpha.\,\alpha) \to \gamma = \bot \to \gamma \tag{5.32}$$

follows. On the other hand we find that

$$(\mu\alpha.\,\alpha) \to \gamma \quad \not\twoheadleftrightarrow_{\mathrm{r/o\text{-}u}(\mu\bot)'} \quad \bot \to \gamma \tag{5.33}$$

is the case: this follows from the observation that the only recursive types ρ that are convertible to $\bot \to \gamma$ with respect to $\twoheadleftrightarrow_{\mathrm{r/o\text{-}u}(\mu\bot)'}$ are of the form

$$\mu\alpha_1 \ldots \alpha_n.\, (\bot \to \gamma)$$

for some $n \in \omega$ and $\alpha_1, \ldots, \alpha_n \in TVar$ such that $\alpha_1, \ldots, \alpha_n \not\equiv \gamma$. But (5.32) and (5.33) obviously contradict Theorem 5.3.7. Hence we must conclude that our assumption, ARROW being correct for $\mathbf{AK}^{=}$, cannot be sustained.

□

Chapter 6

A Duality between $\mathbf{AK}_{\mathbf{0}}^{=}$ and $\mathbf{HB}_{\mathbf{0}}^{=}$

In this chapter a very near correspondence is established between derivations in the variant Brandt-Henglein system $\mathbf{HB}_{\mathbf{0}}^{=}$ and certain specially defined assemblages of derivations, which will be called "consistency-unfoldings", in the variant-syntactic-matching system $\mathbf{AK}_{\mathbf{0}}^{=}$. Because of its immediate, indeed geometric character, we will speak of a *duality*. The idea for this correspondence is due to J.W. Klop who observed a striking similarity between the activities (a) of trying to demonstrate the consistency of an equation with respect to a syntactic-matching system comparable to those in Section 5.2, and (b) of trying to find a derivation for the same equation in a coinductive proof system similar to that of Brandt and Henglein. It turned out, however, that the interconnection suggested by this observation lends itself much better to being formulated as a link between 'normalized' versions of respective syntactic-matching and coinductive proof systems, rather than as a link between respective kinds of proof systems of equational logic containing symmetry and transitivity rules.

This chapter is basically an extended and refined version of both the the paper [Gra02b] and the report [Gra02c]. There, the existence of a "duality" is demonstrated for proof systems concerned with the restriction of recursive type equality $=_{\mu}$ to the set *can-*μTp of recursive types *in canonical form*, whereas here we prove this result for analogous, and more general, proof systems for recursive type equality $=_{\mu}$ on the set μTp of *all* recursive types according to Definition 3.1.1. Apart from this also a number of proofs are given in greater detail here. There is, however, one topic treated in the mentioned papers that is not covered here: in Section 8 of [Gra02b] and [Gra02c], respectively, a sketch is given of how the obtained duality result linking proof systems for recursive type equality can be transferred to a similar pair of proof systems for bisimulation equivalence on equational specifications of cyclic term graphs.

We continue by giving an overview of this chapter. In Section 6.1, we set out to explain the mentioned basic observation, underlying the results in this chapter, for the here relevant case of the 'normalized' versions $\mathbf{AK}_{\mathbf{0}}^{=}$ and $\mathbf{HB}_{\mathbf{0}}^{=}$ of the proof systems $\mathbf{AK}^{=}$ and $\mathbf{HB}^{=}$ for recursive type equality. We demonstrate in an example that a derivation in $\mathbf{HB}_{\mathbf{0}}^{=}$ without open assumptions and with conclusion $\tau = \sigma$ is indeed closely related to the "reflection" of a certain downwards-growing derivation-tree in $\mathbf{AK}_{\mathbf{0}}^{=}$ which formalizes a "successful consistency-check" for the equation between recursive types $\tau = \sigma$ with respect to $\mathbf{AK}_{\mathbf{0}}^{=}$; and that also a similar connection holds via a "reflection" into the opposite direction.

For the purpose of formulating the basic idea illustrated by this example into a precise statement, we introduce three kinds of auxiliary concepts. Firstly, in Section 6.2 we define an extension of the system $\mathbf{HB}_{\mathbf{0}}^{=}$ by adding three more coinductive inference rules that facilitate additional derivations, but that do not lead to more theorems. Secondly, in Section 6.3 we develop a number of notions regarding downwards-growing derivation-trees in $\mathbf{AK}_{\mathbf{0}}^{=}$: finite trees of consequences (f.t.o.c.'s), f.t.o.c.'s with marked formulas and with back-bound leaf-occurrences of marked formulas, partial consistency-unfoldings and consistency unfoldings in $\mathbf{AK}_{\mathbf{0}}^{=}$. And thirdly, in Section 6.4 we define two reflection functions between derivations in the extension of $\mathbf{HB}_{\mathbf{0}}^{=}$ and f.t.o.c.'s with back-bound leaf-occurrences of marked formulas in $\mathbf{AK}_{\mathbf{0}}^{=}$.

Relying on these concepts, our main theorem is then stated and proved in Section 6.5: it asserts that there exists a duality between derivations without open assumption classes in the considered extension of $\mathbf{HB}_{\mathbf{0}}^{=}$ and consistency-unfoldings in $\mathbf{AK}_{\mathbf{0}}^{=}$ via the reflection functions defined in Section 6.4. And we demonstrate by an example that this relationship can geometrically be visualized. In Section 6.6 we investigate the special case of the duality concerning derivations in the basic, not extended, system $\mathbf{HB}_{\mathbf{0}}^{=}$. We show the existence of an analogous duality between derivations without open assumptions in the system $\mathbf{HB}_{\mathbf{0}}^{=}$ (not in its extension) and consistency-unfoldings of a certain formally characterized kind in $\mathbf{AK}_{\mathbf{0}}^{=}$.

In Section 6.7 we gather remarks on our proofs and on the relevance of the duality results. And in particular, we use our main duality theorem to give an alternative soundness proof for the 'normalized' version $\mathbf{HB}_{\mathbf{0}}^{=}$ of the Brandt-Henglein system $\mathbf{HB}^{=}$ that proceeds by 'reducing' the soundness of $\mathbf{HB}_{\mathbf{0}}^{=}$ to the soundness of $\mathbf{AK}_{\mathbf{0}}^{=}$ by applying the duality result.

6.1 The Basic Observation

The results in this chapter have been stimulated by an observation mentioned above of J.W. Klop that indicated a link between the activities of proving the consistency of an equation with respect to a syntactic-matching proof system and of proving an equation in a coinductive proof system like that of Brandt and Henglein.[1] This observation was originally aimed at trying to give an easy argument for the soundness

[1] J.W. Klop made this observation with respect to a pair of proof systems for the binary relation on μ-terms over an arbitrary signature that relates two such terms whenever they have the same tree unfolding.

of a rule similar to the somewhat strange rule ARROW/FIX in the axiomatization of strong recursive type equivalence by Brandt and Henglein, which rule allows applications that formalize a certain form of circular reasoning.

This rule may indeed seem quite paradoxical, at least at first sight: By every application ι of ARROW/FIX in a derivation $\mathcal{D}$ in $\mathbf{HB}^{=}$ or in $\mathbf{HB}^{=}_{\mathbf{0}}$ a formula A of the form $\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2$ is deduced that is also allowed to appear as an assumption in one or in both of the subderivations $\mathcal{D}_1$ and $\mathcal{D}_2$ leading up to the right premise $\tau_1 = \sigma_1$, and respectively, to the left premise $\tau_2 = \sigma_2$ of ι. But furthermore, the conclusion A of ι does *not* depend any more on such assumptions of A in $\mathcal{D}_1$ and $\mathcal{D}_2$ that are discharged by ι. One might initially be left wondering why unsound circular reasoning is avoided by applications of this rule—as this is in fact the case because of the soundness of $\mathbf{HB}^{=}$ and $\mathbf{HB}^{=}_{\mathbf{0}}$ with respect to $=_\mu$ (cf. Theorem 5.1.12 and Theorem 5.1.20). And on the other hand, one might also want to gain a somewhat clearer understanding of how circular reasoning either of $\mathbf{HB}^{=}$ or $\mathbf{HB}^{=}_{\mathbf{0}}$ is actually employed by applications of ARROW/FIX to derive equations between recursive types that lie beyond the power of weak recursive type equivalence $=_{w\mu}$—as this follows from the completeness of $\mathbf{HB}^{=}$ and $\mathbf{HB}^{=}_{\mathbf{0}}$ with respect to $=_\mu$ (cf. again Theorem 5.1.12 and Theorem 5.1.20).

The mentioned observation made it possible to obtain some additional insight into this phenomenon. It pointed to a similarity between the activities of performing a *consistency-check* by *loop checking* with respect to a syntactic-matching system, like $\mathbf{AK}^{=}$, for a given equation, and of trying to construct in bottom-up direction a formal proof for the same equation in a coinductive proof system. Formulated with respect to the systems $\mathbf{AK}^{=}$ and $\mathbf{HB}^{=}$, the original idea consisted of roughly the following elements:

(i) The essential part of a proof for the consistency of an equation $\tau = \sigma$ with respect to the syntactic-matching system $\mathbf{AK}^{=}$ consists in showing that no *such* derivation $\mathcal{D}$ in $\mathbf{AK}^{=}$ leads to a contradiction with respect to $=_\mu$ *that* has only the single assumption $\tau = \sigma$ and in which applications of rules SYMM, TRANS and REN are exclusively used for the purpose of unfolding a recursive type on either side of an equation.

(ii) A systematic overview over all *such* derivations in $\mathbf{AK}^{=}$ from an equation $\tau = \sigma$ can be obtained by developing them simultaneously via stepwise, ‘fair’ extensions of a downwards-growing derivation-tree. Branches are extended as long as no mismatch in leading symbols nor ‘looping’ has been encountered. In this way a *consistency-check* with two possible outcomes can be carried out:

- If $\tau = \sigma$ is not $\mathbf{AK}^{=}$-consistent, then an equation $\tilde{\tau} = \tilde{\sigma}$ is encountered in one of the leaves at the bottom of the derivation-tree such that $\mathcal{L}'(\tilde{\tau}) \neq \mathcal{L}'(\tilde{\sigma})$ holds, i.e. the leading symbols of $\tilde{\tau}$ and $\tilde{\sigma}$ differ, and such that the derivation-tree cannot be extended further from this leaf onwards. In this case the consistency-check fails. Furthermore, from the reached derivation-tree a derivation in $\mathbf{AK}^{=}$ can be extracted that shows that the contradiction $\tilde{\tau} = \tilde{\sigma}$ with respect to $=_\mu$ can be derived in $\mathbf{AK^=}$ from the single assumption $\tau = \sigma$.

- If $\tau = \sigma$ is $\mathbf{AK^=}$-consistent, then no mismatch in leading symbols can be detected during extensions of this derivation-tree. However, also in this case the consistency-check can be completed after finitely many extension-steps since then 'looping' must be encountered in all branches at some stage of the stepwise extension process. Here the consistency-check succeeds with the result of a derivation-tree that can be viewed as a 'witness' for the $\mathbf{AK^=}$-consistency of $\tau = \sigma$ (in the sense that it enables an easy proof to this effect).

(iii) And finally there is the at first somewhat surprising observation: derivation-trees that are the outcome of a successful consistency-check with respect to $\mathbf{AK_0^=}$ can be "reflected", almost simply geometrically, into derivations without open assumptions in the Brandt-Henglein system $\mathbf{HB^=}$.

The aim of trying to simplify this observation as well as of formulating it into a precise statement has provided us with the first and foremost reason for introducing the 'normalized' versions $\mathbf{AK_0^=}$ of $\mathbf{AK^=}$ and $\mathbf{HB_0^=}$ of $\mathbf{HB^=}$. The use of these variant systems has certainly three advantages for describing the ideas explained above: Firstly, in the absence of transitivity and symmetry rules in $\mathbf{AK_0^=}$ the somewhat technical justification of part (i) turns out to be unnecessary, or rather irrelevant, for showing consistency of formulas with respect to $\mathbf{AK_0^=}$. Secondly, due to the special way how the inference rules of $\mathbf{AK_0^=}$ have been chosen, steps in the extensions of derivation-trees for consistency-checks as in (ii) above are of the form of applications of rules of $\mathbf{AK_0^=}$. And thirdly, due to the particular way in which the inference rules of $\mathbf{HB_0^=}$ have been defined in relation to those of $\mathbf{AK_0^=}$, it is the case that most rule applications in $\mathbf{AK_0^=}$ can be reflected directly into rule applications of $\mathbf{HB_0^=}$ (see Section 6.4 for details); this contributes importantly, as we will see later, to making the formal description of part (iii) a much easier matter.

We are now going to explain the observation outlined above, and in particular, its relevant parts (ii) and (iii), in more detail with respect to the proof systems $\mathbf{AK_0^=}$ and $\mathbf{HB_0^=}$. In Example 6.1.1 below, we demonstrate, for two particular strongly equivalent recursive types τ and σ, that a consistency-check for the equation $\tau = \sigma$ with respect to $\mathbf{AK_0^=}$ can be organized as a search until 'looping' occurs for possible mismatches in leading symbols through a downwards-growing "tree of consequences" of $\tau = \sigma$. This consistency-check eventually succeeds by detecting loops in all branches of a finite downwards-growing derivation-tree $\mathcal{C}$, which enables to give an easy inductive proof for the consistency of $\tau = \sigma$ with respect to $\mathbf{AK_0^=}$. Later we will see that $\mathcal{C}$ can be "reflected upwards" in an almost simply geometrical way into a prooftree that is very near to a $\mathbf{HB_0^=}$-derivation $\mathcal{D}$ of $\tau = \sigma$ without open assumptions that is described in Example 6.1.2 below. And furthermore, we will show that the derivation $\mathcal{D}$ can be "reflected downwards", in an even more immediate way, into a derivation-tree that can also be viewed as the outcome of a successful consistency-check with respect to $\mathbf{AK_0^=}$ and that is closely related to $\mathcal{C}$.

We start by giving the mentioned example of a "derivation-tree" in $\mathbf{AK}_{\mathbf{0}}^{=}$ that can be viewed as the outcome of a successful consistency-check and that allows us to give an inductive proof for the consistency with respect to $\mathbf{AK}_{\mathbf{0}}^{=}$ of the equation at its root.

Example 6.1.1 (A successful consistency-check with respect to $\mathbf{AK}_{\mathbf{0}}^{=}$). We consider the two recursive types

$$\tau \equiv \mu\alpha.\,((\alpha \to \alpha) \to \alpha) \qquad \text{and} \qquad \sigma \equiv \mu\alpha.\,(\alpha \to (\alpha \to \alpha))\ , \tag{6.1}$$

which we have encountered before as the recursive types τ_2 and σ_2 in[2] Example 3.6.3. There we have informally convinced ourselves that these recursive types are strongly equivalent.

Now we want to establish $\tau =_\mu \sigma$ in a different manner, namely, by showing that the equation $\tau = \sigma$ is $\mathbf{AK}_{\mathbf{0}}^{=}$-consistent (which is sufficient due to Theorem 5.2.13). For this aim, we consider the assemblage of essentially six different derivations in $\mathbf{AK}_{\mathbf{0}}^{=}$ from the equation $\tau = \sigma$ to the result of the finite downwards-growing *derivation-tree* $\mathcal{C}$ in Figure 6.1. Let this derivation-tree, which we will later call a "finite tree of consequences from $\tau = \sigma$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ with marked formulas", be denoted by $\mathcal{C}$ here. All rule applications indicated by a single or a double ordinary line in $\mathcal{C}$ are applications of rules UNFOLD$_l$ or UNFOLD$_r$ (clearly the single occurrence of a double line in $\mathcal{C}$ indicates the application of both a rule UNFOLD$_l$ and a rule UNFOLD$_r$). The branchings in $\mathcal{C}$ indicated by dashed lines are due to the two possible outcomes of applications of the decomposition-rule DECOMP: a formula $\tilde{\tau}_1 \to \tilde{\tau}_2 = \tilde{\sigma}_1 \to \tilde{\sigma}_2$ gives rise to a branching indicated by a dashed line below this formula and with the formulas $\tilde{\tau}_1 = \tilde{\sigma}_1$ and $\tilde{\tau}_2 = \tilde{\sigma}_2$ as the left, and respectively, as the right successor of $\tilde{\tau}_1 \to \tilde{\tau}_2 = \tilde{\sigma}_1 \to \tilde{\sigma}_2$ in the $\mathcal{C}$. The meaning of the superscript-markers u, v and w for some formulas in $\mathcal{C}$ will be explained below.

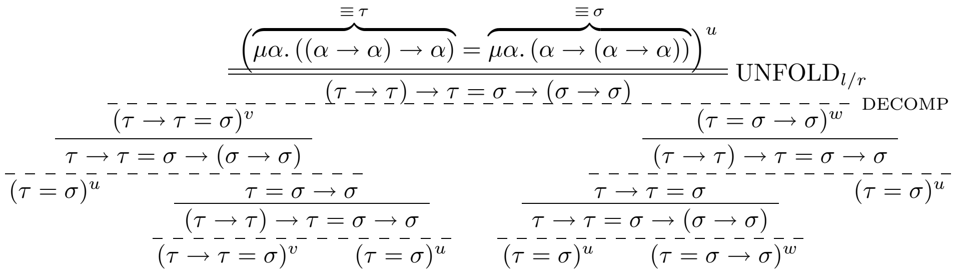
This derivation-tree $\mathcal{C}$ has a particular property, which can be used for proving the consistency of $\tau = \sigma$ with respect to $\mathbf{AK}_{\mathbf{0}}^{=}$. For explaining this feature of $\mathcal{C}$ we neglect, for the moment, the fact that the rule REN is part of the system $\mathbf{AK}_{\mathbf{0}}^{=}$ (in the presence of this rule it has to be argued in a somewhat more complicated way[3]). In doing so and by furthermore neglecting the two possibilities for the order in which the applications of UNFOLD$_l$ and of UNFOLD$_r$ can be performed in the first two applications of a REN-free derivation from $\tau = \sigma$ in $\mathbf{AK}_{\mathbf{0}}^{=}$, we can state the mentioned distinctive property of $\mathcal{C}$ as follows: $\mathcal{C}$ assembles and displays in a graphic way all six possibilities for derivations from the equation $\tau = \sigma$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ up to that particular point in each derivation, when for the first time a formula is derived that has been encountered before. The markers u, v and w have been used to indicate at which earlier occurrences the formulas at the bottom of $\mathcal{C}$ have respectively appeared in $\mathcal{C}$ for the first time.

Each of the six mentioned derivations gathered in $\mathcal{C}$ correspond to threads in $\mathcal{C}$ from the root to a leaf at its bottom. We denote these derivations respectively by

[2] With the intention of avoiding potentially distracting subscripts as much as possible, we use different names for these recursive types here.

[3] Cf. the proof of the implication of "⇐" of Theorem 6.3.18 below.

Figure 6.1: Assemblage to a finite downwards-growing “tree of consequences” $\mathcal{C}$ of the six different possible derivations in $\mathbf{AK}_{\mathbf{0}}^{=}$ without REN-applications from the assumption $\mu\alpha.\,((\alpha \to \alpha) \to \alpha) = \mu\alpha.\,(\alpha \to (\alpha \to \alpha))$ with the additional property that the derivation ends as soon as *looping* is encountered for the first time.


$\equiv \tau$
$\equiv \sigma$
$\left(\mu\alpha.\,((\alpha \to \alpha) \to \alpha) = \mu\alpha.\,(\alpha \to (\alpha \to \alpha))\right)^u$
UNFOLD$_{l/r}$
$(\tau \to \tau) \to \tau = \sigma \to (\sigma \to \sigma)$
DECOMP
$(\tau \to \tau = \sigma)^v$
$(\tau = \sigma \to \sigma)^w$
$\tau \to \tau = \sigma \to (\sigma \to \sigma)$
$(\tau \to \tau) \to \tau = \sigma \to \sigma$
$(\tau = \sigma)^u$
$\tau = \sigma \to \sigma$
$\tau \to \tau = \sigma$
$(\tau = \sigma)^u$
$(\tau \to \tau) \to \tau = \sigma \to \sigma$
$\tau \to \tau = \sigma \to (\sigma \to \sigma)$
$(\tau \to \tau = \sigma)^v$
$(\tau = \sigma)^u$
$(\tau = \sigma)^u$
$(\tau = \sigma \to \sigma)^w$

$\mathcal{D}_1, \ldots, \mathcal{D}_6$ according to the order in which the corresponding threads are encountered in $\mathcal{C}$ while passing from left to right (we recall the imprecision of this statement resulting from the fact that every thread from the root to a leaf in $\mathcal{C}$ stands actually for two derivations in $\mathbf{AK}_{\mathbf{0}}^{=}$ because no order has been fixed in the symbolic prooftree $\mathcal{C}$ for the two topmost applications of UNFOLD$_l$ and UNFOLD$_r$). Hence, for instance, $\mathcal{D}_5$ is a derivation in $\mathbf{AK}_{\mathbf{0}}^{=}$ from assumption $\tau = \sigma$ and with conclusion $\tau = \sigma \to \sigma$; furthermore, $\mathcal{D}_5$ has depth 7 and consists of applications of UNFOLD$_{l/r}$, DECOMP, UNFOLD$_l$, DECOMP, UNFOLD$_r$ and DECOMP in this order from its single assumption to the conclusion. We say that $\mathcal{D}_1, \ldots, \mathcal{D}_6$ *span* the derivation-tree $\mathcal{C}$ shown in in Figure 6.1.

Now we are going to use the distinctive feature of $\mathcal{C}$ explained above for showing that no contradiction with respect to $=_\mu$ can be derived from $\tau = \sigma$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ by an $\mathbf{AK}_{\mathbf{0}}^{=}$-derivation without REN-applications. More precisely, we will show that

$$\left.\begin{array}{l}\text{For all derivations } \mathcal{D} \text{ in } \mathbf{AK}_{\mathbf{0}}^{=} \text{ from the assumption } \tau = \sigma \text{ without}\\ \text{REN-applications and with conclusion } \tilde{\tau} = \tilde{\sigma} \text{ (for some } \tilde{\tau}, \tilde{\sigma} \in \mu Tp)\\ \text{it holds that } \mathcal{L}'(\tilde{\tau}) = \mathcal{L}'(\tilde{\sigma}) \text{, i.e. that the conclusion of } \mathcal{D} \text{ is no}\\ \text{contradiction with respect to } =_\mu .\end{array}\right\} \qquad (6.2)$$

holds. We will do so by induction on the depth $n = |\mathcal{D}|$ of derivations $\mathcal{D}$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ from $\tau = \sigma$. Thereby we will rely on the important observation

$$(\forall\, \tilde{\tau} = \tilde{\sigma} \text{ equation in the deriv.-tree } \mathcal{C} \text{ in Fig. 6.1}) \left[\, \mathcal{L}'(\tilde{\tau}) = \to = \mathcal{L}'(\tilde{\sigma}) \,\right], \qquad (6.3)$$

which asserts in particular that the derivation-tree $\mathcal{C}$ in Figure 6.1 does not contain contradictions with respect to $=_\mu$. Assertion (6.3) can easily be checked.

For settling the base case of the induction for (6.2), we notice that τ and σ have the same leading symbol, namely $\to$ (we have already observed this before as part of (6.3)).

For the treatment of the induction step for (6.2), we assume the induction hypothesis and let $\mathcal{D}$ be an arbitrary derivation in $\mathbf{AK}_{\mathbf{0}}^{=}$ from the assumption $\tau = \sigma$

with depth $|\mathcal{D}| = n+1 \geq 1$; we will show that the conclusion of $\mathcal{D}$ is no contradiction with respect to $=_\mu$.

We distinguish the cases $|\mathcal{D}| = n+1 \leq 7$ and $|\mathcal{D}| = n+1 > 7$. In the first case we will demonstrate the assertion needed for the induction step directly, that is, without using the induction hypothesis, whereas we will rely on the induction hypothesis for settling the second case.

For the case $|\mathcal{D}| = n+1 \leq 7$, we treat the two subcases $1 \leq |\mathcal{D}| = n+1 \leq 5$ and $5 < |\mathcal{D}| = n+1 \leq 7$ separately.

In the first subcase $1 \leq |\mathcal{D}| = n+1 \leq 5$, the derivation $\mathcal{D}$ must be an initial segment of one of the derivations $\mathcal{D}_1, \ldots, \mathcal{D}_6$ that span the derivation-tree $\mathcal{C}$ because these six derivations comprise all possible initial-segments of REN-free $\mathbf{AK}_{\mathbf{0}}^{=}$-derivations from $\tau = \sigma$ until looping occurs and all of these derivations contain at least 5 rule applications. This implies that the conclusion of $\mathcal{D}$ is contained in $\mathcal{C}$. Hence observation (6.3) implies now that the conclusion of $\mathcal{D}$ is not a contradiction with respect to $=_\mu$.

In the second subcase $5 < |\mathcal{D}| = n+1 \leq 7$, it follows similarly due to the construction of $\mathcal{C}$ that either $\mathcal{D}$ is an initial segment of one of the four derivations $\mathcal{D}_2, \mathcal{D}_3, \mathcal{D}_4$ or $\mathcal{D}_5$ of depth 7 contained in $\mathcal{C}$, or that $\mathcal{D}$ has one of the derivations $\mathcal{D}_1$ or $\mathcal{D}_6$ in $\mathcal{C}$ of depth 5 as initial-segment. In the first situation it is obvious that the conclusion of $\mathcal{D}$ is contained in $\mathcal{C}$, and hence (6.3) implies that the conclusion of $\mathcal{D}$ is not a contradiction with respect to $=_\mu$. In the second situation, however, $\mathcal{D}$ must contain a loop consisting of 5 rule applications, more precisely, the assumption $\tau = \sigma$ is reached again in $\mathcal{D}$ after 5 rule applications below the assumption $\tau = \sigma$. By removing this loop from $\mathcal{D}$, a derivation $\mathcal{D}_0$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ from the assumption $\tau = \sigma$ results that has depth $1 \leq |\mathcal{D}_0| \leq 2$ and the same conclusion as $\mathcal{D}$. From what we have shown in the first subcase above, we can now conclude that the conclusion of $\mathcal{D}_0$, and hence the conclusion of $\mathcal{D}$, is not a contradiction with respect to $=_\mu$. Hereby we have shown the induction step for the second subcase.

In this way we have successfully performed the induction step for the first case, in which $|\mathcal{D}| = n+1 \leq 7$ holds (as noted earlier, we did not use the induction hypothesis in this case).

In the second case, where $|\mathcal{D}| = n+1 > 7$ holds, it follows that precisely one of the derivations $\mathcal{D}_1, \ldots, \mathcal{D}_6$ spanning the derivation-tree $\mathcal{C}$ is an initial-segment of $\mathcal{D}$. Since each of $\mathcal{D}_1, \ldots, \mathcal{D}_6$ contains a loop of at least four rule applications, it is possible to remove at least four rule applications from $\mathcal{D}$ with the result of a derivation $\mathcal{D}_0$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ from the assumption $\tau = \sigma$ that has the same conclusion as $\mathcal{D}$ and that has depth $|\mathcal{D}_0| \leq |\mathcal{D}| - 4$. By applying the induction hypothesis for $\mathcal{D}_0$ it follows that $\mathcal{D}_0$ does not have a contradiction with respect to $=_\mu$ as its conclusion. And this entails that $\mathcal{D}$ cannot have a contradiction with respect to $=_\mu$ as its conclusion, neither.

Hereby we have performed the induction step also for the case $|\mathcal{D}| = n+1 > 7$. Consequently, we have succeeded in showing (6.2).

Thus we have given an argument, the refinement of which (with respect to the presence of REN-rules in $\mathbf{AK}_{\mathbf{0}}^{=}$) can indeed show that $\tau = \sigma$ is consistent with $\mathbf{AK}_{\mathbf{0}}^{=}$. Our proof, as well as its possible refinement, is based on the derivation-tree

$\mathcal{C}$ in $\mathbf{AK}_{\mathbf{0}}^{=}$, the outcome of a successful consistency-check for the equation $\tau = \sigma$ by *loop-checking* for all (REN-free) derivations from $\tau = \sigma$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ and by arranging these derivations to a downwards-growing derivation-tree. The derivation-tree $\mathcal{C}$ can be viewed as a 'witness' for the $\mathbf{AK}_{\mathbf{0}}^{=}$-consistency of the equation $\tau = \sigma$ at its root, in the sense that it enables an easy proof by induction, of the kind described above, for the impossibility to derive a contradiction with respect to $=_\mu$ from $\tau = \sigma$ in $\mathbf{AK}_{\mathbf{0}}^{=}$. Such derivation-trees will later be called "consistency-unfoldings in $\mathbf{AK}_{\mathbf{0}}^{=}$" with respect to a notion that will formally be introduced in Section 6.3. ■

We continue by giving a formal derivation in the system $\mathbf{HB}_{\mathbf{0}}^{=}$ for the equation $\tau = \sigma$ with the recursive types τ and σ as in the above example.

Example 6.1.2 (A formal proof in $\mathbf{HB}_{\mathbf{0}}^{=}$). Let τ and σ be the recursive types considered in Example 6.1.1 that are specified in (6.1). In Example 3.6.3 we saw that $\tau =_\mu \sigma$ holds. Due to completeness of the system $\mathbf{HB}_{\mathbf{0}}^{=}$ with respect to $=_\mu$, there must then exist a proof of $\tau = \sigma$ in $\mathbf{HB}_{\mathbf{0}}^{=}$. If one goes about to build such a proof in bottom-up direction and thereby avoids unnecessary applications of the rules REN of taking variants, and always uses the rule ARROW/FIX as soon as possible to discharge arising assumptions, then there is essentially only one possibility for such a proof. Namely, in this way the derivation $\mathcal{D}$ shown in Figure 6.2 in the system $\mathbf{HB}_{\mathbf{0}}^{=}$ is found. Each of the three applications of two-premise rules in $\mathcal{D}$ that are respectively labeled by u, v and w are applications of the rule ARROW/FIX at which the assumptions in $\mathcal{D}$ marked by u, v or w are discharged respectively. ■

Now it is very striking to notice, for τ and σ as in Example 6.1.1 and in Example 6.1.2, a direct relationship between the outcome of the successful consistency-check for $\tau = \sigma$ in $\mathbf{AK}_{\mathbf{0}}^{=}$, the derivation-tree $\mathcal{C}$ in Figure 6.1, and the derivation $\mathcal{D}$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ of $\tau = \sigma$ in Figure 6.2. Namely, each of $\mathcal{C}$ and $\mathcal{D}$ is, if some details are overlooked for the moment, very close to the 'mirror image' of the other.

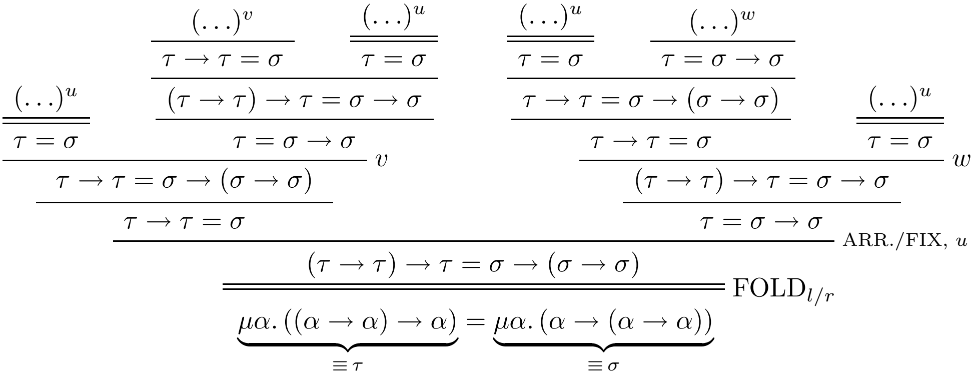
Let us explain this for the derivation $\mathcal{D}$ first. By reflecting the downwards-growing derivation-tree $\mathcal{C}$ from Figure 6.1 at a horizontal line, the upwards-growing derivation-tree Refl($\mathcal{C}$) in Figure 6.3 is reached. During this reflection all applications of UNFOLD$_l$ and UNFOLD$_r$ in $\mathcal{C}$ are changed into respective applications of FOLD$_l$ and FOLD$_r$, and all branchings DECOMP are replaced by applications of ARROW. This prooftree Refl($\mathcal{C}$) is essentially a derivation in $\mathbf{HB}_{\mathbf{0}}^{=}$ that has occurrences of undischarged marked assumptions at its top, but that contains, unlike $\mathbf{HB}_{\mathbf{0}}^{=}$-derivations, still some marked formulas in its 'interior'. However, Refl($\mathcal{C}$) is very near to $\mathcal{D}$ although it possesses open assumptions, while $\mathcal{D}$ does not. Closer inspection shows the following: if Refl($\mathcal{C}$) is transformed further

(1) by extending Refl($\mathcal{C}$) above all of its leaves appropriately by one or by two applications of FOLD$_l$ and/or FOLD$_r$,

(2) by transferring the respective assumption markers in Refl($\mathcal{C}$) up to the formulas in the new leaves, and

(3) by discharging these newly arising assumptions at respective applications of ARROW deeper down in Refl($\mathcal{C}$), thereby removing the respective assumption

Figure 6.2: The derivation $\mathcal{D}$ of $\mu\alpha.\,((\alpha \to \alpha) \to \alpha) = \mu\alpha.\,(\alpha \to (\alpha \to \alpha))$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ without open assumption classes.

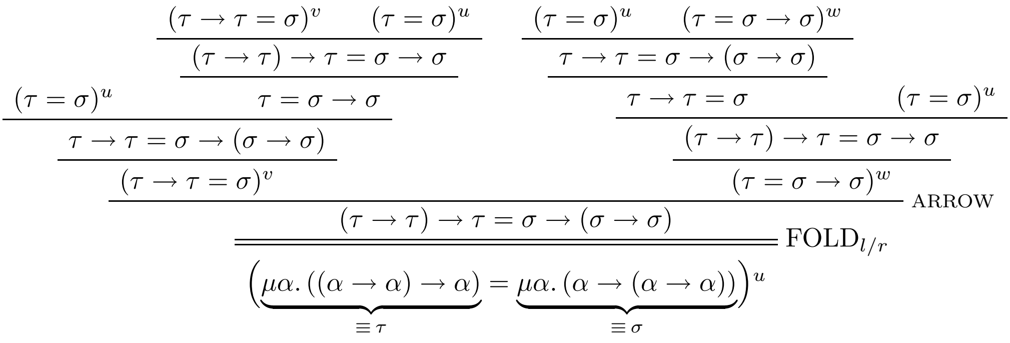


$(\dots)^v$
$(\dots)^u$
$(\dots)^u$
$(\dots)^w$
$\tau \to \tau = \sigma$
$\tau = \sigma$
$\tau = \sigma$
$\tau = \sigma \to \sigma$
$(\dots)^u$
$(\tau \to \tau) \to \tau = \sigma \to \sigma$
$\tau \to \tau = \sigma \to (\sigma \to \sigma)$
$(\dots)^u$
$\tau = \sigma$
$\tau = \sigma \to \sigma$
$\tau \to \tau = \sigma$
$\tau = \sigma$
v
w
$\tau \to \tau = \sigma \to (\sigma \to \sigma)$
$(\tau \to \tau) \to \tau = \sigma \to \sigma$
$\tau \to \tau = \sigma$
$\tau = \sigma \to \sigma$
ARR./FIX, u
$(\tau \to \tau) \to \tau = \sigma \to (\sigma \to \sigma)$
FOLD$_{l/r}$
$\underbrace{\mu\alpha.\,((\alpha \to \alpha) \to \alpha)}_{\equiv\tau} = \underbrace{\mu\alpha.\,(\alpha \to (\alpha \to \alpha))}_{\equiv\sigma}$

For typographical reasons the marked assumptions at the top of the derivation $\mathcal{D}$ have been abbreviated as follows:

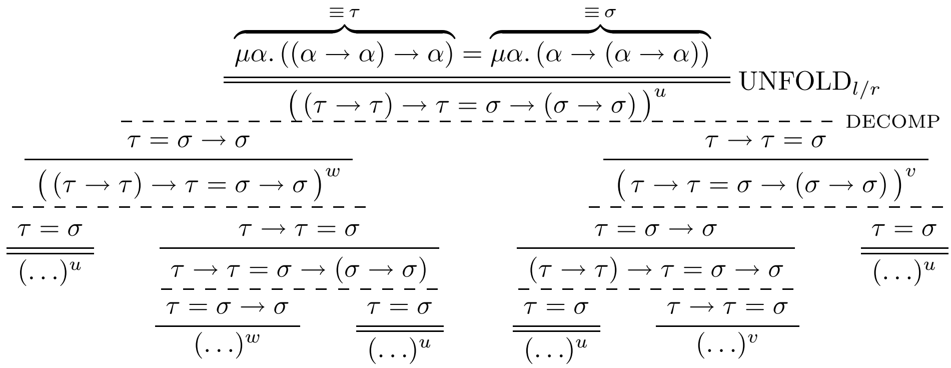
$$\begin{array}{lll}
(\dots)^u & \text{stands for} & (\,(\tau \to \tau) \to \tau) = \sigma \to (\sigma \to \sigma)\,)^u \quad , \\
(\dots)^v & \qquad\quad \text{for} & (\,\tau \to \tau = \sigma \to (\sigma \to \sigma)\,)^v \quad \text{and} \\
(\dots)^w & \qquad\quad \text{for} & (\,(\tau \to \tau) \to \tau = \sigma \to \sigma\,)^w \quad .
\end{array}$$

Figure 6.3: Derivation Refl($\mathcal{C}$) in $\mathbf{HB}_{\mathbf{0}}^{=}$ that is the result of a reflection of the derivation-tree $\mathcal{C}$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ from Figure 6.1 at a horizontal line, during which DECOMP-branchings are mirrored into ARROW-applications and UNFOLD$_{l/r}$- into FOLD$_{l/r}$-applications.



$(\tau \to \tau = \sigma)^v$
$(\tau = \sigma)^u$
$(\tau = \sigma)^u$
$(\tau = \sigma \to \sigma)^w$
$(\tau \to \tau) \to \tau = \sigma \to \sigma$
$\tau \to \tau = \sigma \to (\sigma \to \sigma)$
$(\tau = \sigma)^u$
$\tau = \sigma \to \sigma$
$\tau \to \tau = \sigma$
$(\tau = \sigma)^u$
$\tau \to \tau = \sigma \to (\sigma \to \sigma)$
$(\tau \to \tau) \to \tau = \sigma \to \sigma$
$(\tau \to \tau = \sigma)^v$
$(\tau = \sigma \to \sigma)^w$
ARROW
$(\tau \to \tau) \to \tau = \sigma \to (\sigma \to \sigma)$
FOLD$_{l/r}$
$\Big(\underbrace{\mu\alpha.\,((\alpha \to \alpha) \to \alpha)}_{\equiv\tau} = \underbrace{\mu\alpha.\,(\alpha \to (\alpha \to \alpha))}_{\equiv\sigma}\Big)^u$

Figure 6.4: Derivation-tree Refl($\mathcal{D}$) in $\mathbf{AK}_{\mathbf{0}}^{=}$ that is the result of a reflection of the derivation $\mathcal{D}$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ from Figure 6.2: ARROW/FIX-applications are mirrored into DECOMP-branchings and FOLD$_{l/r}$- into UNFOLD$_{l/r}$-applications, and furthermore, the reflections of conclusion-formulas of ARROW/FIX-applications in $\mathcal{D}$ are marked according to the respectively discharged assumptions.


$\overbrace{\mu\alpha.\,((\alpha \to \alpha) \to \alpha)}^{\equiv\,\tau} = \overbrace{\mu\alpha.\,(\alpha \to (\alpha \to \alpha))}^{\equiv\,\sigma}$
UNFOLD$_{l/r}$
$\big((\tau \to \tau) \to \tau = \sigma \to (\sigma \to \sigma)\big)^{u}$
DECOMP
$\tau = \sigma \to \sigma$
$\tau \to \tau = \sigma$
$\big((\tau \to \tau) \to \tau = \sigma \to \sigma\big)^{w}$
$\big(\tau \to \tau = \sigma \to (\sigma \to \sigma)\big)^{v}$
$\tau = \sigma$
$\tau \to \tau = \sigma$
$\tau = \sigma \to \sigma$
$\tau = \sigma$
$(\ldots)^{u}$
$\tau \to \tau = \sigma \to (\sigma \to \sigma)$
$(\tau \to \tau) \to \tau = \sigma \to \sigma$
$(\ldots)^{u}$
$\tau = \sigma \to \sigma$
$\tau = \sigma$
$\tau = \sigma$
$\tau \to \tau = \sigma$
$(\ldots)^{w}$
$(\ldots)^{u}$
$(\ldots)^{u}$
$(\ldots)^{v}$

(For the leaves of Refl($\mathcal{D}$) we have used, again for typographical reasons, the abbreviations explained in Figure 6.2).

marker $\boldsymbol{u}$ in the conclusions of these ARROW-applications and changing the application itself to an application of ARROW/FIX at which $\boldsymbol{u}$ is discharged,

then a derivation Refl($\mathcal{C}$)* in $\mathbf{HB}_{\mathbf{0}}^{=}$ is built that is identical to $\mathcal{D}$. In this way, we have recognized that $\mathcal{D}$ is indeed closely related to a 'mirror image' of $\mathcal{C}$.

And analogously, it can easily be seen that the reflection Refl($\mathcal{D}$) shown in Figure 6.4 of the derivation $\mathcal{D}$ in Figure 6.2 can be viewed as formalizing the outcome of a successful consistency-check for the equation $\tau = \sigma$ in $\mathbf{AK}_{\mathbf{0}}^{=}$. For carrying out this reflection, we first 'pre-process' $\mathcal{D}$ by transferring markers from assumptions that are discharged at ARROW/FIX-applications to the conclusions of the respective ARROW/FIX-application; let us denote the resulting prooftree by $\mathcal{D}^{(m)}$. Only then is $\mathcal{D}^{(m)}$ mirrored downwards at a horizontal line with the result Refl($\mathcal{D}$) of a downwards-growing derivation-tree in $\mathbf{AK}_{\mathbf{0}}^{=}$ from the assumption $\tau = \sigma$. During this reflection ARROW/FIX-applications are mirrored into DECOMP-branchings, and FOLD$_{l/r}$- into UNFOLD$_{l/r}$-applications.

Then it is easy to see that a very similar proof as given in Example 6.1.1 for the consistency with respect to $\mathbf{AK}_{\mathbf{0}}^{=}$ of the formula $\tau = \sigma$ at the root of $\mathcal{C}$ can also be given using the derivation-tree Refl($\mathcal{D}$) instead of using $\mathcal{C}$. Thus also Refl($\mathcal{C}$) can be viewed as a derivation-tree in $\mathbf{AK}_{\mathbf{0}}^{=}$ that formalizes a successful consistency-check for $\tau = \sigma$. And furthermore, Refl($\mathcal{D}$) is very close to the derivation-tree $\mathcal{C}$, more precisely, it contains respectively only one or two applications of UNFOLD$_l$ and/or UNFOLD$_r$ below each leaf of $\mathcal{C}$, and as a further difference with $\mathcal{C}$, and the backbinding markers are organized slightly in a slightly different manner (for example, no backbinding occurs in Refl($\mathcal{D}$) to the formula at the root). But we find nevertheless, that the derivation-tree $\mathcal{C}$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ is closely related to a 'mirror image' of the $\mathbf{HB}_{\mathbf{0}}^{=}$-derivation $\mathcal{D}$.

These observations do indeed suggest a very close relationship between derivation-trees in $\mathbf{AK}_{\mathbf{0}}^{=}$ that formalize successful consistency-checks in $\mathbf{AK}_{\mathbf{0}}^{=}$ and derivations in $\mathbf{HB}_{\mathbf{0}}^{=}$ without open assumptions. We will formalize this relationship as a precise proof-theoretic interconnection between an extension of $\mathbf{HB}_{\mathbf{0}}^{=}$ and the system $\mathbf{AK}_{\mathbf{0}}^{=}$ in the course of this chapter.

There is one further hint to be taken from our examples above for the development of notions needed for the formalization of this observation in the coming sections. Seemingly, derivation-trees in $\mathbf{AK}_{\mathbf{0}}^{=}$ that can be used to show the consistency with respect to $\mathbf{AK}_{\mathbf{0}}^{=}$ of the equation at their root are slightly more general formal objects, when compared via reflection operations, than prooftrees of derivations in $\mathbf{HB}_{\mathbf{0}}^{=}$ without open assumptions. As an indication for this we have noticed that the reflection Refl($\mathcal{D}$) of the derivation $\mathcal{D}$ in Figure 6.2 can directly be considered to be the outcome of a successful consistency-check in $\mathbf{AK}_{\mathbf{0}}^{=}$, whereas vice versa, the reflection Refl($\mathcal{C}$) of the derivation-tree $\mathcal{C}$ in Figure 6.1 had to be 'post-processed' to reach a derivation in $\mathbf{HB}_{\mathbf{0}}^{=}$ without open assumptions (by adding one or two additional rule applications above each marked assumption at the top of Refl($\mathcal{C}$), and by arranging new bindings to ARROW/FIX-applications for the purpose of discharging the newly created open assumptions there). Hence a general concept of backbinding for leaf-occurrences of marked formulas in $\mathbf{AK}_{\mathbf{0}}^{=}$-derivation trees might be stronger, when compared via appropriate reflection operations, than the concept of discharging assumptions in $\mathbf{HB}_{\mathbf{0}}^{=}$-derivations.

From this the question arises whether, and if so then how, the system $\mathbf{HB}_{\mathbf{0}}^{=}$ can be extended in such a way that the mechanism for discharging assumptions in the extended system is equally strong as a general concept of backbinding for marked assumptions in $\mathbf{AK}_{\mathbf{0}}^{=}$-derivation-trees as used above. Our duality results will ultimately show that the answer is yes for the particular extension of $\mathbf{HB}_{\mathbf{0}}^{=}$ that is introduced in the next section as well as for formalizations of successful consistency-checks with respect to $\mathbf{AK}_{\mathbf{0}}^{=}$ by "consistency-unfoldings" in Section 6.3.

6.2 The Extension e-$\mathbf{HB}_{\mathbf{0}}^{=}$ of $\mathbf{HB}_{\mathbf{0}}^{=}$

For the purpose of formulating the observation described in Section 6.1 into a 'smooth' formal relationship between the systems $\mathbf{AK}_{\mathbf{0}}^{=}$ and $\mathbf{HB}_{\mathbf{0}}^{=}$, it will turn out to be useful to extend the system $\mathbf{HB}_{\mathbf{0}}^{=}$. As indicated at the end of the previous section, in particular the introduction of such additional rules will be of significance that allow assumptions to be discharged in situations in which the rule ARROW/FIX of $\mathbf{HB}_{\mathbf{0}}^{=}$ is not applicable. We are therefore going to enrich $\mathbf{HB}_{\mathbf{0}}^{=}$ by a number of such rules of this kind that facilitate additional derivations, but that do not cause new theorems to become derivable in the extended system (in view of the completeness of $\mathbf{HB}_{\mathbf{0}}^{=}$ with respect to $=_{\mu}$, an extension of $\mathbf{HB}_{\mathbf{0}}^{=}$ that increased the set of theorems would clearly be undesirable).

Before introducing these new rules, we define and study a generalization of the rule ARROW/FIX, the rule FIX, that will subsequently help us to show that the introduced additional rules are (cr-)admissible in $\mathbf{HB}_{\mathbf{0}}^{=}$. The rule FIX is an inference rule with identical premise and conclusion that allows to discharge a formula of the same form as its premise and conclusion; but applications of FIX are only allowed, if a condition on the "contractiveness" of the derivation leading to its premise is satisfied.

Definition 6.2.1 (Contractiveness of derivations, the rule FIX). Let $\mathcal{S}$ be a natural-deduction system whose set of formulas is the set μTp–Eq of equations between recursive types and whose rules include the rules ARROW and ARROW/FIX of $\mathbf{HB}_{\mathbf{0}}^{=}$.

(i) Let $\mathcal{D}$ be a derivation in $\mathcal{S}$, and let $\tau, \sigma \in \mu Tp$ and u be an assumption marker. We say that $\mathcal{D}$ is *contractive with respect to open marked assumptions* $(\tau = \sigma)^u$ if and only if the following holds:

> For every undischarged marked assumption of the form $(\tau = \sigma)^u$ in $\mathcal{D}$, the thread down to the conclusion of $\mathcal{D}$ *crosses* an application of ARROW or ARROW/FIX *at least once*. (6.4)

(ii) The rule FIX is an inference rule whose applications at the bottom of derivations $\mathcal{D}$ in $\mathcal{S}$+FIX are of the form

$$\dfrac{\begin{matrix}[\tau = \sigma]^u \\ \mathcal{D}_1 \\ \tau = \sigma\end{matrix}}{\tau = \sigma}\,\text{FIX}, u \tag{6.5}$$

(with some $\tau, \sigma \in \mu Tp$ and an assumption marker u), where the immediate subderivation $\mathcal{D}_1$ of $\mathcal{D}$ is a derivation in $\mathcal{S}$+FIX that is *contractive* with respect to open marked assumptions $(\tau = \sigma)^u$.

⊠

Remark 6.2.2. (a) If the system $\mathcal{S}$ in Definition 6.2.1 is taken to be the system $\mathbf{HB}_{\mathbf{0}}^{=}$, then due to the fact that the only two-premise rules in $\mathbf{HB}_{\mathbf{0}}^{=}$ are the rules ARROW and ARROW/FIX, condition (6.4), which defines when a derivation $\mathcal{D}$ is contractive with respect to open marked assumptions $(\tau = \sigma)^u$, can easily be seen to be equivalent to the following easier, and more concrete formulation:

> There are *either no* undischarged marked assumptions in $\mathcal{D}$ of the form $(\tau = \sigma)^u$ *or* there is *at least one* occurrence of an application of ARROW or ARROW/FIX in $\mathcal{D}$. (6.6)

Our preference for (6.4) over (6.6) as the defining clause for the notion of "contractiveness" of a derivation $\mathcal{D}$ with respect to marked open assumptions

$(\tau = \sigma)^u$ is motivated by the following fact. The definition of the rule FIX in Definition 6.2.1, (b), with the side-condition that the immediate subderivation of a FIX-application is "contractive" with respect to the discharged assumptions in the sense of Definition 6.2.1, (i), turns out to be the 'right' one also for the situation when FIX is considered in the context of the basic, not-analytic, Brandt-Henglein system $\mathbf{HB}^=$ (see the results reported in Chapter 9, Section 9.2, Subsection 9.2.1); this would not be the case if in Definition 6.2.1, (i), the clause (6.6) were chosen instead of (6.4).

(b) The rule FIX does not conform fully to the format of "ANDS-rule" as introduced in Definition B.2.1 in Appendix B. This is because for checking whether an inference labeled by FIX in a given derivation is a (formally) correct application of FIX not only the conclusion, the premise, the rule name label, and the open assumptions belonging to this application have to be looked at; it also has to be verified that the application's immediate subderivation is indeed contractive with respect to the open marked assumptions (of the form of the premise and the conclusion that get discharged). This is not the case for derivations in the concept of ANDS as developed in Section B.2 of Appendix B.

A possibility to formulate a version of the rule FIX that is formalizable as an ANDS-rule exists in an annotated version **ann-**$\mathbf{HB}_0^=$ of $\mathbf{HB}_0^=$, which is defined in Definition 8.1.1, Chapter 8. This system has the property that *some* information about the structure of a derivation $\mathcal{D}$ in **ann-**$\mathbf{HB}_0^=$ is contained in the annotation χ within the conclusion $\chi : \tau = \sigma$ of $\mathcal{D}$. This feature of the system **ann-**$\mathbf{HB}_0^=$ makes it possible to express the condition that a derivation $\mathcal{D}$ in **ann-**$\mathbf{HB}_0^=$ is contractive with respect to certain open assumptions as a condition about the annotation in the conclusion of $\mathcal{D}$. As a consequence, a version of the rule FIX can be defined in an extension of the system **ann-**$\mathbf{HB}_0^=$ such that this version of FIX can be formalized as an ANDS-rule (see Remark 8.1.2, (b), in Chapter 8).

It turns out that the addition of the rule FIX to the system $\mathbf{HB}_0^=$ does not lead to the consequence that additional theorems would become derivable. In fact, as this can be seen from the proof of the following lemma, the 'deductive power' of a particular application of FIX in a derivation $\mathcal{D}$ in $\mathbf{HB}_0^=$+FIX can always be emulated by the 'deductive power' of a certain application[4] of ARROW/FIX in a derivation $\mathcal{D}'$ that is closely related to $\mathcal{D}$.

Lemma 6.2.3. *The rule* FIX *is a cr-admissible rule in* $\mathbf{HB}_0^=$*, which can be eliminated effectively from arbitrary derivations in* $\mathbf{HB}_0^=$+FIX*: every derivation* $\mathcal{D}$ *in* $\mathbf{HB}_0^=$+FIX *can effectively be transformed into a derivation* $\mathcal{D}'$ *in* $\mathbf{HB}_0^=$ *with the same conclusion and the same (if any) open assumption classes.*

[4] This application of ARROW/FIX is either already present in $\mathcal{D}$ or it arises from an application of ARROW present in $\mathcal{D}$ (cf. the proof of Lemma 6.2.3).

Proof. It suffices to show that applications of the rule FIX can effectively be eliminated from arbitrary derivations $\mathcal{D}$ of the form (6.5), where $\mathcal{D}_1$ is a derivation in $\mathbf{HB}_{\mathbf{0}}^{=}$ that is contractive with respect to open marked assumptions of the form $(\tau = \sigma)^u$, with the result of a derivation $\mathcal{D}'$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ that has the same conclusion, the same (if any) open assumption classes, and that furthermore satisfies the condition:

$$\left.\begin{array}{l} \text{For every } \rho, \chi \in \mu Tp \text{ and assumption markers } v \text{ it holds that:} \\ \quad \mathcal{D} \text{ is contractive w.r.t. open marked assumptions } (\rho = \chi)^v \quad \Longrightarrow \\ \qquad \Longrightarrow \quad \mathcal{D}' \text{ is contractive w.r.t. open marked assumptions } (\rho = \chi)^v \ . \end{array}\right\} \quad (6.7)$$

(We therefore have to show slightly more than cr-correctness of FIX in $\mathbf{HB}_{\mathbf{0}}^{=}$: we have to demonstrate the fulfilledness of the defining clause of cr-admissibility for FIX in $\mathbf{HB}_{\mathbf{0}}^{=}$, and to make sure additionally that the transformation can be carried out effectively and that for its result (6.7) holds.) If this has been proven, it follows (a) that appropriate eliminations of topmost occurrences of FIX in an arbitrary derivation $\mathcal{D}$ in $\mathbf{HB}_{\mathbf{0}}^{=}$+FIX can be carried out effectively without affecting the side-conditions of other applications of FIX in $\mathcal{D}$, and hence (b) that all applications of FIX in an arbitrary derivation $\mathcal{D}$ in $\mathbf{HB}_{\mathbf{0}}^{=}$+FIX can be removed effectively by successive eliminations of always topmost occurrences first with the desired result of a derivation $\mathcal{D}'$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ that has the same conclusion and the same open assumption classes as $\mathcal{D}$. (The argument sketched here is analogous to how the implication "cr-correctness ⇒ cr-admissibility" in Lemma 4.3.5 can be shown.)

We therefore let an arbitrary derivation $\mathcal{D}$ of the form (6.5) be given (with some $\tau, \sigma \in \mu Tp$), for which we assume that $\mathcal{D}_1$ is a derivation in $\mathbf{HB}_{\mathbf{0}}^{=}$ that is contractive with respect to open marked assumptions $(\tau = \sigma)^u$. We want to show that $\mathcal{D}$ can effectively be transformed into a derivation $\mathcal{D}'$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ with the same conclusion, the same (if any) open assumption classes, and for which furthermore (6.7) is true.

In case that the open assumption class $[\tau = \sigma]^u$ in $\mathcal{D}_1$ is empty, the application of FIX at the bottom of $\mathcal{D}$ amounts to a trivial step, in which no assumption is discharged and no change of the equation between recursive types in the premise occurs. Hence this application of FIX can be removed from $\mathcal{D}$ and we can take $\mathcal{D}'$ to be simply $\mathcal{D}_1$.

Now we assume that there is at least one open marked assumption $(\tau = \sigma)^u$ in $\mathcal{D}_1$. From the fact that $\mathcal{D}_1$ is contractive with respect to such open assumptions we know that the thread from each such assumption downwards in $\mathcal{D}_1$ to the conclusion $\tau = \sigma$ crosses an application of ARROW or ARROW/FIX at least once. Hence there is at least one application of ARROW or ARROW/FIX in $\mathcal{D}_1$.[5] We will now consider the bottommost such application in $\mathcal{D}_1$, with respect to which then all

[5] We could also have concluded this by using (6.6).

marked assumptions of $\mathcal{D}_1$ must be located above it. Thus $\mathcal{D}_1$ can be written as

$$\begin{array}{cl} [\tau = \sigma]^u & \\ \mathcal{D}_{1a} & \\ \text{- - - - - - - -} & \text{ARROW or ARROW/FIX}, v \\ (\tau_{1a} = \sigma_{1a}) & \\ \mathcal{D}_1 & \\ \tau = \sigma & \end{array}$$

for some $\tau_{1a}, \sigma_{1a} \in \mu Tp$, a derivation $\mathcal{D}_{1a}$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ (with open assumptions) and a derivation $\mathcal{D}_1$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ with a single marked assumption $(\tau_{1a} = \sigma_{1a})^w$ (for some assumption marker w) that does *only contain* applications of one-premise rules. Thereby the bottommost application of ARROW or ARROW/FIX, v in $\mathcal{D}_1$ is indicated by the dotted line that represents the last rule application in $\mathcal{D}_{1a}$, which derivation still stretches across this application and has conclusion $\tau_{1a} = \sigma_{1a}$.

The open assumption class $[\tau = \sigma]^u$ in $\mathcal{D}_1$ (and in $\mathcal{D}_{1a}$) can now be "closed" by inserting the derivation $\mathcal{D}_{1a}$ in $\mathcal{D}_1$ above all open marked assumptions $(\tau = \sigma)^u$ of this class and by then discharging the newly arising assumption class $(\tau_{1a} = \sigma_{1a})^w$ at the location of the bottommost application of ARROW or ARROW/FIX in $\mathcal{D}_1$: in case that this displayed last rule application in $\mathcal{D}_{1a}$ is an application of ARROW, it is changed to an application of (ARROW/FIX, v) and the marker w for the assumption class $[\tau_{1a} = \sigma_{1a}]^w$ in the derivation $\mathcal{D}_1$ is at the same time changed from w to v for a new marker v, which does not occur in $\mathcal{D}_1$; in case this displayed application in $\mathcal{D}_{1a}$ is an application (ARROW/FIX, v) only the marker w for the assumption class $[\tau_{1a} = \sigma_{1a}]^w$ in $\mathcal{D}_1$ is changed to v. – This means that in both cases $\mathcal{D}_1$ is transformed to the $\mathbf{HB}_{\mathbf{0}}^{=}$-derivation $\mathcal{D}'$

$$\begin{array}{cl} (\tau_{1a} = \sigma_{1a})^v & \\ \mathcal{D}_1 & \\ [\tau = \sigma] & \\ \mathcal{D}_{1a} & \\ \text{- - - - - - - -} & \text{ARROW/FIX}, v \\ (\tau_{1a} = \sigma_{1a}) & \\ \mathcal{D}_1 & \\ \tau = \sigma & \end{array}$$

which possesses the same open assumption classes as $\mathcal{D}_1$ except for $[\tau = \sigma]^u$ and thus has the same conclusion and the same open assumption classes as $\mathcal{D}$. Furthermore also (6.7) holds, since in this case clearly both $\mathcal{D}$ and $\mathcal{D}'$ are contractive with respect to arbitrary open marked assumptions. Hence the result $\mathcal{D}'$ of effectively eliminating the application FIX from the given derivation $\mathcal{D}$ of the form (6.5) is in fact of the required form.

□

Now we proceed to give the extension mentioned above of the system $\mathbf{HB}_{\mathbf{0}}^{=}$ by adding new rules. We extend $\mathbf{HB}_{\mathbf{0}}^{=}$ by three rules, each of which is closely related to a respective one-premise rule of $\mathbf{HB}_\mathbf{0}^=$, but differs from that by the property that all of its applications discharge one or more present marked assumptions.

Definition 6.2.4 (The extension e-$\mathbf{HB}_\mathbf{0}^=$ of the system $\mathbf{HB}_\mathbf{0}^=$). The extension **e-**$\mathbf{HB}_\mathbf{0}^=$ of the system $\mathbf{HB}_\mathbf{0}^=$ has the same formulas and axioms as $\mathbf{HB}_\mathbf{0}^=$, allows to make the same marked assumptions and contains all inference rules of $\mathbf{HB}_\mathbf{0}^=$. Additionally, **e-**$\mathbf{HB}_\mathbf{0}^=$ contains the rules REN/FIX, FOLD$_l$/FIX and FOLD$_r$/FIX defined below.

Applications of these rules arise from applications of, respectively, the rules REN, FOLD$_l$ or FOLD$_r$ by the stipulation that at least one marked assumptions of the form of the respective conclusion is discharged. More precisely, applications in **e-**$\mathbf{HB}_\mathbf{0}^=$ of REN/FIX and of FOLD$_l$/FIX together with their immediate subderivations $\mathcal{D}_1$ in **e-**$\mathbf{HB}_\mathbf{0}^=$ are of the respective forms

$$\dfrac{\begin{matrix}[\tau' = \sigma']^u \\ \mathcal{D}_1 \\ \tau = \sigma\end{matrix}}{\tau' = \sigma'}\ \text{REN/FIX},\, u \qquad \begin{array}{l}(\text{where } \tau' \equiv_{\text{ren}} \tau \text{ and } \sigma' \equiv_{\text{ren}} \sigma \\ \quad \text{and if side-conditions } \mathbf{C} \text{ and } \mathbf{I})\end{array} \tag{6.8}$$

(with some $\tau, \sigma, \tau', \sigma' \in \mu Tp$ and an assumption marker u) and

$$\dfrac{\begin{matrix}\big[\,\mu\alpha.\,\tau = \sigma\big]^u \\ \mathcal{D}_1 \\ \tau[\mu\alpha.\,\tau/\alpha] = \sigma\end{matrix}}{\mu\alpha.\,\tau = \sigma}\ \text{FOLD}_l/\text{FIX},\, u \qquad (\text{if side-conditions } \mathbf{C} \text{ and } \mathbf{I}) \tag{6.9}$$

(with some $\tau, \sigma \in \mu Tp$ and $\alpha \in TVar$ and an assumption marker u), where the side-conditions **C** and **I** on the derivations $\mathcal{D}_1$ are defined as follows. The side-condition **C** asserts in each case that the derivation $\mathcal{D}_1$ is contractive (cf. Definition 6.2.1 (i)) with respect to open marked assumptions with assumption marker u (which assumptions are discharged at the respective application of REN/FIX or of FOLD$_l$/FIX). And side-condition **I** asserts that at least one open marked assumption is discharged by the respective application of REN/FIX or of FOLD$_l$/FIX. Thus in (6.8) the side-condition **I** demands that the open assumption class $[\tau' = \sigma']^u$ is inhabited, whereas in (6.9) it requires that the open assumption class $[\mu\alpha.\,\tau = \sigma]^u$ is inhabited).

The rule FOLD$_r$/FIX that has an analogous relationship with the rule FOLD$_r$ of $\mathbf{HB}_\mathbf{0}^=$ as FOLD$_l$/FIX has with FOLD$_l$ is defined similarly; in particular, applications of FOLD$_r$/FIX are subject to analogous side-conditions **C** and **I**.

As an abbreviation used in informal arguments, we will again allow to speak about one or about both of the rules FOLD$_l$/FIX and FOLD$_r$/FIX by using the expression FOLD$_{l/r}$/FIX.

☒

Remark 6.2.5. (a) The side-condition **I** on the new rules in **e-**$\mathbf{HB}_\mathbf{0}^=$ has been taken up for the same reason why it has been required also earlier in the definition of the system $\mathbf{HB}_{\mathbf{0}}^{=}$ (contrasting with the rule ARROW/FIX in the 'original' Brandt-Henglein system $\mathbf{HB}^{=}$ for which it is not demanded): to create a clear notational distinction between rules, applications of which cause some present assumptions to be discharged, and rules, applications of which do not discharge any assumptions. As a consequence, no application of one of the rules $\text{FOLD}_{l/r}$, REN or ARROW in an **e-**$\mathbf{HB}_{\mathbf{0}}^{=}$-derivation can be viewed as just a special case of an application of $\text{FOLD}_{l/r}$/FIX, REN/FIX or ARROW/FIX. The aim here is to make it possible to discriminate between, on the one hand, applications of $\text{FOLD}_{l/r}$, REN and ARROW, and on the other hand, applications of $\text{FOLD}_{l/r}$/FIX, REN/FIX and ARROW/FIX for the purpose of easing the formal reasoning about later defined transformations of derivations.

(b) That the side-condition **C** on subderivations leading up to applications of one of the new rules in **e-**$\mathbf{HB}_{\mathbf{0}}^{=}$ is essential can easily be seen. If it were not demanded, then equations $\tau = \sigma$ that equate recursive types that are not strongly equivalent would become derivable. This is obvious for the rule REN/FIX: for any $\tau, \sigma \in \mu Tp$ such that $\tau \neq_{\mu} \sigma$, consider the derivation with conclusion $\tau = \sigma$ that consists just of one (trivial) application of REN/FIX at which a marked assumption $(\tau = \sigma)^{u}$ (for some marker u) in the premise is discharged. And in the case of the rule FOLD_{l}/FIX, consider, for arbitrary $\alpha, \beta, \alpha_1, \alpha_2 \in TVar$ such that $\alpha_1 \not\equiv \alpha_2$, the derivation

$$\cfrac{\cfrac{(\,\mu\alpha_1\alpha_2.\,\alpha_1 \,=\, \alpha \to \beta\,)^{u}}{\mu\alpha_2\alpha_1\alpha_2.\,\alpha_1 \,=\, \alpha \to \beta}\ \text{FOLD}_{l}}{\mu\alpha_1\alpha_2.\,\alpha_1 \,=\, \alpha \to \beta}\ \text{FOLD}_{l}/\text{FIX},\, u$$

without open assumptions; obviously an analogous derivation can be given using the rules FOLD_{r} and FOLD_{r}/FIX. Therefore a system with rules $\text{FOLD}_{l/r}$/ FIX and REN/FIX for applications of which the side-condition **C** were not demanded would be unsound with respect to strong recursive type equivalence.

(c) Similarly as the system $\mathbf{HB}_{\mathbf{0}}^{=}$ possesses a FIX-*pendant* to the rule ARROW in the rule ARROW/FIX, the system **e-**$\mathbf{HB}_{\mathbf{0}}^{=}$ contains FIX-pendants to the rules $\text{FOLD}_{l/r}$ and REN of $\mathbf{HB}_{\mathbf{0}}^{=}$. In connection with this it could be asked why in **e-**$\mathbf{HB}_{\mathbf{0}}^{=}$ no FIX-pendants have been defined for the rules $(\mu-\bot)_{l/r}^{\bot\text{der}}$ of $\mathbf{HB}_{\mathbf{0}}^{=}$.

A first observation related to the reason for this is the following. No derivation $\mathcal{D}_1$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ or in **e-**$\mathbf{HB}_{\mathbf{0}}^{=}$ with the conclusion in, for example, the premise $\bot = \sigma$ of a rule $(\mu-\bot)_{l}^{\bot\text{der}}$ contains an application of ARROW or ARROW/FIX (because otherwise, as one can easily verify, both recursive types in the conclusion of $\mathcal{D}_1$ would contain the symbol $\to$). Therefore no such derivation $\mathcal{D}_1$ can be contractive. And furthermore, a hypothetical rule

$(\mu\text{–}\bot)_l^{\bot\text{der}}$/FIX with applications of the form

$$\frac{\begin{array}{c}[\mu\alpha\alpha_1\ldots\alpha_n.\,\alpha=\sigma]^u\\ \mathcal{D}_1\\ \bot=\sigma\end{array}}{\mu\alpha\alpha_1\ldots\alpha_n.\,\alpha=\sigma}\;(\mu\text{–}\bot)_l^{\bot\text{der}}/\text{FIX},\,u\quad\text{(if side-conditions }\mathbf{C}\text{ and }\mathbf{I}\text{)}$$

could never be applied, even if the side-condition **C** were dropped. This is because there does not exist a derivation in $\mathbf{HB}_{\mathbf{0}}^{=}$, nor in $\mathbf{e\text{-}HB}_{\mathbf{0}}^{=}$, that contains (as required by the side-condition **I**) an occurrence of $(\bot=\sigma)^u$ as an open marked assumption (for some marker u) and that has the conclusion $\mu\alpha\alpha_1\ldots\alpha_n.\,\alpha=\sigma$.

On the other hand, a hypothetical rule $\tilde{R}$ with applications of the form

$$\frac{\begin{array}{c}[\bot=\sigma]^u\\ \mathcal{D}_1\\ \bot=\sigma\end{array}}{\mu\alpha\alpha_1\ldots\alpha_n.\,\alpha=\sigma}\;\tilde{R},\,u\quad\text{(if side-condition }\mathbf{I}\text{)}$$

in which the side-condition **C** on $\tilde{R}$ were not demanded, would obviously make it possible to derive contradictions with respect to $=_\mu$ as theorems (such as, for example, the contradiction $\mu\alpha.\,\alpha=\mu\alpha.\,(\alpha\to\alpha)$ with respect to $=_\mu$).

We will now see that, as a consequence of the fact that FIX is cr-admissible in $\mathbf{HB}_{\mathbf{0}}^{=}$, also the additional rules in $\mathbf{e\text{-}HB}_{\mathbf{0}}^{=}$ are cr-admissible in $\mathbf{HB}_{\mathbf{0}}^{=}$, and that the systems $\mathbf{e\text{-}HB}_{\mathbf{0}}^{=}$ and $\mathbf{HB}_{\mathbf{0}}^{=}$ are equivalent.

Theorem 6.2.6 (Equivalence of the systems $\mathbf{HB}_{\mathbf{0}}^{=}$ and $\mathbf{e\text{-}HB}_{\mathbf{0}}^{=}$). *The following three statements hold about the relationship between the systems* $\mathbf{HB}_{\mathbf{0}}^{=}$ *and* $\mathbf{e\text{-}HB}_{\mathbf{0}}^{=}$*:*

(i) Every derivation $\mathcal{D}$ in $\mathbf{e\text{-}HB}_{\mathbf{0}}^{=}$ *can effectively be transformed into a derivation $\mathcal{D}'$ in* $\mathbf{HB}_{\mathbf{0}}^{=}$ *with the same conclusion and the same (if any) open assumption classes.*

(ii) The rules REN/FIX, FOLD$_l$/FIX *and* FOLD$_r$/FIX *are cr-admissible rules in* $\mathbf{HB}_{\mathbf{0}}^{=}$.

(iii) The system $\mathbf{e\text{-}HB}_{\mathbf{0}}^{=}$ *is a conservative extension of* $\mathbf{HB}_{\mathbf{0}}^{=}$*, and hence*[6] *the systems* $\mathbf{HB}_{\mathbf{0}}^{=}$ *and* $\mathbf{e\text{-}HB}_{\mathbf{0}}^{=}$ *are equivalent (i.e. they possess the same theorems).*

Proof. Since both statements (ii) and (iii) of the theorem are obvious consequences of statement (i), we only have to show item (i) of the theorem.

For this notice first that an arbitrary application of one of the rules REN/FIX and FOLD$_{l/r}$/FIX in a derivation in $\mathbf{e\text{-}HB}_{\mathbf{0}}^{=}$+FIX can be eliminated effectively by

[6] Since $\mathbf{HB}_{\mathbf{0}}^{=}$ and $\mathbf{e\text{-}HB}_{\mathbf{0}}^{=}$ have the same formulas.

making use of a respective application of the rule FIX. Let $\mathcal{D}$ be an arbitrary derivation in **e-**$\mathbf{HB}_{\mathbf{0}}^{=}$+FIX. Then it holds that each application of the rule FOLD$_l$/FIX in $\mathcal{D}$ can be replaced by a succession of an application of FOLD$_l$ with an application of FIX. More precisely, every subderivation $\mathcal{D}_1$ of $\mathcal{D}$ of the form

$$\begin{array}{c}\big[\,\mu\alpha.\,\tau=\sigma\,\big]^{u}\\ \mathcal{D}_1\\ \dfrac{\tau[\mu\alpha.\,\tau/\alpha]=\sigma}{\mu\alpha.\,\tau=\sigma}\;\mathrm{FOLD}_l/\mathrm{FIX},\,u\end{array}\qquad\text{can be replaced by}\qquad\begin{array}{c}\big[\,\mu\alpha.\,\tau=\sigma\,\big]^{u}\\ \mathcal{D}_1\\ \dfrac{\dfrac{\tau[\mu\alpha.\,\tau/\alpha]=\sigma}{\mu\alpha.\,\tau=\sigma}\;\mathrm{FOLD}_l}{\mu\alpha.\,\tau=\sigma}\;\mathrm{FIX},\,u\end{array}$$

(which derivation we call $\mathcal{D}_1$) in $\mathcal{D}$ with the result of a derivation $\mathcal{D}^{(1)}$. The side-condition **C** on the application of FOLD$_l$/FIX at the bottom of $\mathcal{D}_1$ in $\mathcal{D}$ implies thereby the side-condition on the new application of FIX in $\mathcal{D}_1$ and in $\mathcal{D}^{(1)}$. It is obvious that this replacement of an application of FOLD$_l$/FIX does not affect the validity of the side-conditions of any other application of FIX or of R/FIX for $R \in \{\mathrm{FOLD}_{l/r}, \mathrm{REN}\}$ in (the transformation-step from $\mathcal{D}$ to) $\mathcal{D}^{(1)}$. Furthermore—because clearly $\mathcal{D}_1$ and $\mathcal{D}_1$ have the same open assumption classes—also the derivations $\mathcal{D}$ and $\mathcal{D}^{(1)}$ possess the same open assumption classes; but $\mathcal{D}^{(1)}$ has one application of (a rule REN/FIX, FOLD$_r$/FIX or) FOLD$_l$/FIX less than $\mathcal{D}$. Analogous transformation-steps can be enacted for arbitrary applications of FOLD$_r$/FIX and of REN/FIX in $\mathcal{D}$.

For showing (i), we let $\mathcal{D}$ be an arbitrary derivation in **e-**$\mathbf{HB}_{\mathbf{0}}^{=}$. We will show that $\mathcal{D}$ can effectively be transformed into a derivation $\mathcal{D}'$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ with the same conclusion and the same open assumption classes. By successively eliminating applications of REN/FIX or FOLD$_{l/r}$/FIX in a way as indicated above, $\mathcal{D}$ can effectively be transformed into a derivation $\mathcal{D}^{(f)}$ in $\mathbf{HB}_{\mathbf{0}}^{=}$+FIX with the same conclusion and with the same open assumption classes as $\mathcal{D}$. And furthermore, the derivation $\mathcal{D}^{(f)}$ can be transformed, due to Lemma 6.2.3, in an effective way into a derivation $\mathcal{D}'$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ with the same conclusion and the same (if any) undischarged assumption classes as $\mathcal{D}^{(f)}$ by eliminating all FIX-applications from $\mathcal{D}^{(f)}$. Hence a derivation $\mathcal{D}'$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ that mimics $\mathcal{D}$ can be effectively produced. In this way we have shown statement (i) of the theorem.

□

As an aside, we will prove now that each of the additional rules REN/FIX and FOLD$_{l/r}$/FIX in **e-**$\mathbf{HB}_{\mathbf{0}}^{=}$ can be used as a counterexample for the validity of the implication "admissibility ⇒ derivability" for rules in natural-deduction systems: due to item (ii) of the theorem above and the following proposition, these rules can be used to justify the third sentence in Lemma 4.3.5, (ii).

Proposition 6.2.7. *None of the rules* REN/FIX *or* FOLD$_{l/r}$/FIX *of* **e-**$\mathbf{HB}_{\mathbf{0}}^{=}$ *is a derivable rule of* $\mathbf{HB}_{\mathbf{0}}^{=}$.

Proof. We will only show that FOLD$_l$/FIX is not derivable in $\mathbf{HB}_{\mathbf{0}}^{=}$. For the rules FOLD$_r$/FIX and REN/FIX it can be argued similarly.

For the sake of the argument, we assume that $\mathrm{FOLD}_l/\mathrm{FIX}$ is derivable in $\mathbf{HB}_{\mathbf{0}}^{=}$. We let α and γ be distinct type variables, and we let the recursive type τ be defined as $\tau \equiv \mu\alpha.\,(\alpha \to \gamma)$. Furthermore, we consider an extension by enlargement $\mathcal{S}_{\mathrm{ext}}$ of $\mathbf{HB}_{\mathbf{0}}^{=}$ by adding the rule $\mathrm{FOLD}_l/\mathrm{FIX}$ as well as the rule DECOMP_r that allows applications of the form

$$\dfrac{\begin{array}{c}\mathcal{D}_1\\ \tilde{\tau} = \tilde{\sigma} \to \tilde{\sigma}\end{array}}{\tilde{\tau} = \tilde{\sigma}}\;\mathrm{DECOMP}_r$$

(for all $\tilde{\tau}, \tilde{\sigma} \in \mu Tp$). Clearly, most of the inferences enabled by DECOMP_r are unsound with respect to recursive type equality $=_\mu$.

However, since, according to our assumption, $\mathrm{FOLD}_l/\mathrm{FIX}$ is derivable in $\mathbf{HB}_{\mathbf{0}}^{=}$, it follows: for the derivation $\mathcal{D}$ in $\mathcal{S}_{\mathrm{ext}}$ that is of the form

$$\dfrac{\dfrac{\dfrac{(\tau = \gamma \to \gamma)^u}{\tau = \gamma}\;\mathrm{DECOMP}_r \qquad \dfrac{\scriptstyle(\mathrm{REFL})}{\gamma = \gamma}}{\tau \to \gamma = \gamma \to \gamma}\;\mathrm{ARROW}}{\underbrace{\mu\alpha.\,(\alpha \to \gamma)}_{\equiv\tau} = \gamma \to \gamma}\;\mathrm{FOLD}_l/\mathrm{FIX},\, u \tag{6.10}$$

and that does not contain open assumptions there exists a unary derivation context $\mathcal{DC}'$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ such that the prooftree $\mathcal{D}'$ of the form

$$\begin{array}{c}\dfrac{\dfrac{(\tau = \gamma \to \gamma)^u}{\tau = \gamma}\;\mathrm{DECOMP}_r \qquad \dfrac{\scriptstyle(\mathrm{REFL})}{\gamma = \gamma}}{\big[\tau \to \gamma = \gamma \to \gamma\big]_1}\;\mathrm{ARROW}\\ \mathcal{DC}'\\ \tau = \gamma \to \gamma\end{array} \tag{6.11}$$

is a derivation in $\mathcal{S}_{\mathrm{ext}}$ that mimics $\mathcal{D}$. We choose $\mathcal{DC}'$ and $\mathcal{D}'$ in this way. Since $\mathcal{D}'$ mimics $\mathcal{D}$, also $\mathcal{D}'$ does not contain open assumptions.

We note that $\mathcal{DC}'$ must indeed contain occurrences of the hole $[\,]_1$: otherwise $\mathcal{DC}'$ would be equal to $\mathcal{D}'$, implying that $\mathcal{DC}'$ would be a derivation in $\mathbf{HB}_{\mathbf{0}}^{=}$ ($\mathcal{DC}'$ is a derivation context in $\mathbf{HB}_{\mathbf{0}}^{=}$!) that does not contain open assumptions, and hence the conclusion of $\mathcal{DC}'$ would be a theorem of $\mathbf{HB}_{\mathbf{0}}^{=}$; but this cannot be the case since, due to $\tau \neq_\mu \gamma \to \gamma$, it would contradict Theorem 5.1.20, the soundness of $\mathbf{HB}_{\mathbf{0}}^{=}$ with respect to recursive type equality $=_\mu$.

Hence it follows that the open assumptions $(\tau = \gamma \to \gamma)^u$ indicated at the top of the symbolic prooftree (6.11) for $\mathcal{D}'$ correspond to actual occurrences of marked assumptions in $\mathcal{D}'$. Since $\mathcal{D}'$ does not contain open assumptions, these occurrences of marked assumptions must be discharged in $\mathcal{D}'$ at rule applications within $\mathcal{DC}'$. But this is not possible since $\mathcal{DC}'$ is a derivation context in $\mathbf{HB}_{\mathbf{0}}^{=}$: the single rule of $\mathbf{HB}_{\mathbf{0}}^{=}$ that allows assumptions to be discharged is ARROW/FIX, and this rule only allows equations between *composite* recursive types to be discharged at its applications; the recursive type τ on the left-hand side of the marked assumption $(\tau = \gamma \to \gamma)^u$ is, however, not composite.

Therefore our initial assumption that FOLD$_l$/FIX is a derivable rule in $\mathbf{HB}_{\mathbf{0}}^{=}$ cannot be sustained.

□

This proof for the non-derivability of either of the three additional rules in $\mathbf{HB}_{\mathbf{0}}^{=}$ can be summed up informally as follows: these rules are not derivable in **e-**$\mathbf{HB}_{\mathbf{0}}^{=}$ because their ability to discharge also marked assumptions of the form $(\tau = \sigma)^u$, where at least one of τ or σ is *not* of the form $\chi_1 \to \chi_2$ (for some $\chi_1, \chi_2 \in \mu Tp$) cannot be 'simulated' in $\mathbf{HB}_{\mathbf{0}}^{=}$-derivations, in which only marked assumptions of the form $(\rho_1 \to \rho_2 = \chi_1 \to \chi_2)^v$ (for some $\rho_1, \rho_2, \chi_1, \chi_2 \in \mu Tp$ and markers v) are able to get discharged.

6.3 Consistency-Unfoldings in $\mathbf{AK}_{\mathbf{0}}^{=}$

In this section we are concerned with formalizing downwards-growing derivation-trees in $\mathbf{AK}_{\mathbf{0}}^{=}$ of the kind that we have encountered in Section 6.1. Our aim is to introduce a formal notion for such derivation-trees that can be viewed as 'witnesses' of successful consistency-checks with respect to $\mathbf{AK}_{\mathbf{0}}^{=}$ for the equation at the respective root. We will realize this goal, towards the end of this section, by the definition of "consistency-unfoldings in $\mathbf{AK}_{\mathbf{0}}^{=}$". For this purpose, we need a number of auxiliary notions for derivation-trees in $\mathbf{AK}_{\mathbf{0}}^{=}$ starting with "finite trees of consequences in $\mathbf{AK}_{\mathbf{0}}^{=}$". This basic notion will then be enriched by allowing marked formulas to occur in such trees, and subsequently by a concept of backbinding for leaf-occurrences of marked formulas, leading to the notion of "finite trees of consequences in $\mathbf{AK}_{\mathbf{0}}^{=}$ with back-bound leaf-occurrences of marked formulas". For illustrating these upcoming definitions, both of the downwards-growing derivation-trees in $\mathbf{AK}_{\mathbf{0}}^{=}$ appearing in Section 6.1, $\mathcal{C}$ in Figure 6.1 and Refl($\mathcal{C}$) in Figure 6.4, are going to serve us as running examples.

We start by defining a "finite tree of consequence in $\mathbf{AK}_{\mathbf{0}}^{=}$" $\mathcal{C}$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ from a given equation $\tau = \sigma$ between recursive types as the assemblage of finitely many derivations of finite length from $\tau = \sigma$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ to the form of a derivation-tree; its depth $|\mathcal{C}|$ will be defined as the maximal number of rule applications encountered in one of the $\mathbf{AK}_{\mathbf{0}}^{=}$-derivations that make up the derivation-tree $\mathcal{C}$.

Definition 6.3.1 (Finite trees of consequences in $\mathbf{AK}_{\mathbf{0}}^{=}$). A *finite tree of consequences* (a *f.t.o.c.*) $\mathcal{C}$ of, or from, an equation $\tau = \sigma$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ is an object that together with the assertion "$\mathcal{C}$ is a f.t.o.c. of $\tau = \sigma$ in $\mathbf{AK}_{\mathbf{0}}^{=}$" can be formed by a finite number of applications of the following three generation rules. Thereby also the depth $|\mathcal{C}|$ of $\mathcal{C}$ is defined in parallel with these rules, in which we use auxiliary framed boxes to delimit symbolic denotations for f.t.o.c.'s from the surrounding text:

(i) For all $\tau, \sigma \in \mu Tp$, the recursive type equation $\boxed{\tau = \sigma}$ is a f.t.o.c. $\mathcal{C}$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ from $\tau = \sigma$ in $\mathbf{AK}_{\mathbf{0}}^{=}$. The depth $|\mathcal{C}|$ of $\mathcal{C}$ is defined as $|\mathcal{C}| =_{\text{def}} 0$.

(ii) For all $\tau, \sigma, \tau_1, \sigma_1 \in \mu Tp$, $\boxed{\begin{array}{c} \dfrac{\tau = \sigma}{\tau_1 = \sigma_1}\, R \\ \mathcal{C}_1 \end{array}}$ is a f.t.o.c. $\mathcal{C}$ from $\tau = \sigma$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ given that $\mathcal{C}_1$ is a f.t.o.c. of $\tau_1 = \sigma_1$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ and $\dfrac{\tau = \sigma}{\tau_1 = \sigma_1}\, R$ is a rule application of one of the rules UNFOLD$_l$, UNFOLD$_r$, $(\mu{-}\bot)_l^{\text{der}\bot}$, $(\mu{-}\bot)_r^{\text{der}\bot}$ or REN of $\mathbf{AK}_{\mathbf{0}}^{=}$. Here the depth $|\mathcal{C}|$ of $\mathcal{C}$ is defined as $|\mathcal{C}| =_{\text{def}} |\mathcal{C}_1| + 1$.

(iii) For all $\tau_1, \tau_2, \sigma_1, \sigma_2 \in \mu Tp$, $\boxed{\begin{array}{cc} \multicolumn{2}{c}{\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2} \\ \hdashline \tau_1 = \sigma_1 & \tau_2 = \sigma_2 \\ \mathcal{C}_1 & \mathcal{C}_2 \end{array} \text{ DECOMP}}$ is a f.t.o.c. $\mathcal{C}$ from $\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ given that $\mathcal{C}_1$ is a f.t.o.c. of $\tau_1 = \sigma_1$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ and that $\mathcal{C}_2$ is a f.t.o.c. of $\tau_2 = \sigma_2$ in $\mathbf{AK}_{\mathbf{0}}^{=}$. Here we set $|\mathcal{C}| =_{\text{def}} 1 + \max\{|\mathcal{C}_1|, |\mathcal{C}_2|\}$ for the depth of the f.t.o.c. $\mathcal{C}$. For the part of the derivation-tree $\mathcal{C}$ involving the dashed line, namely for

$$\begin{array}{cc} \multicolumn{2}{c}{\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2} \\ \hdashline \tau_1 = \sigma_1 & \tau_2 = \sigma_2 \end{array} \text{ DECOMP}$$

we will use the term *branching*; and for such a branching we say that it has the formulas $\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2$, $\tau_1 = \sigma_1$ and $\tau_2 = \sigma_2$ as its respective *premise*, *left conclusion* and *right conclusion*.

$\boxtimes$

Example 6.3.2. It is quite obvious to see that both the derivation-trees $\mathcal{C}$ in Figure 6.1 and Refl($\mathcal{D}$) in Figure 6.4 are, if the assumption markers u, v and w are removed that are attached to some equations in $\mathcal{C}$ and in Refl($\mathcal{D}$), f.t.o.c.'s in $\mathbf{AK}_{\mathbf{0}}^{=}$ from the equation

$$\mu\alpha.\,((\alpha \to \alpha) \to \alpha) = \mu\alpha.\,(\alpha \to (\alpha \to \alpha))\ ; \tag{6.12}$$

they have respective depth $|\mathcal{C}| = 7$ and $|\text{Refl}(\mathcal{D})| = 9$. ■

Notation 6.3.3 (Symbolic representation of trees of consequences). For dealing with derivation-trees such as the above defined f.t.o.c.'s in $\mathbf{AK}_{\mathbf{0}}^{=}$ in a practical way, for example in definitions and proofs, we will use a similar notation as for prooftrees. There will actually be a symmetry between our notations for derivation-trees and for prooftrees because the first are downwards-branching whereas the second are upwards-branching trees. Instead of giving a rigorous definition for these symbolic representations of derivation-trees, two examples should suffice to make clear the analogy between our representation of derivation-trees and the notation used for prooftrees explained in Section 2.2 and used everywhere else here.

- We will use a symbolic derivation-tree of the form

$$\begin{array}{c} \tau = \sigma \\ \mathcal{C} \\ (\rho = \chi) \end{array}$$

as denotation for a f.t.o.c. $\mathcal{C}$ from $\tau = \sigma$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ in which we have fixed (by some closer description or assumption in the context of a specific argument) a single existing leaf, which carries the equation $\rho = \chi$, from among the leaves at the bottom of the derivation-tree $\mathcal{C}$.

- A symbolic derivation-tree like

$$\begin{array}{c} \tau = \sigma \\ \mathcal{C}_1 \\ (\rho_1 = \chi_1) \\ \mathcal{C}_2 \\ (\rho_2 = \chi_2) \end{array}$$

will be used for denoting a f.t.o.c. from $\tau = \sigma$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ that is the result of substituting a f.t.o.c. $\mathcal{C}_2$ from $\rho_1 = \chi_1$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ (in which one occurrence of $\rho_2 = \chi_2$ at a leaf of $\mathcal{C}_2$ is symbolically referred to and fixed) into (and this means below) the single fixed and displayed leaf with the equation $\rho_1 = \chi_1$ in a f.t.o.c. $\mathcal{C}_1$ from $\tau = \sigma$ in $\mathbf{AK}_{\mathbf{0}}^{=}$.

This notation for the symbolic treatment of derivation-trees is going to be developed further and extended to "f.t.o.c.'s with marked formulas" and "f.t.o.c.'s with back-bound leaf-occurrences of marked formulas" in the two definitions ahead. ⊠

Definition 6.3.4 (Finite trees of consequences with marked formulas; back-bound and unbound leaf-occurrences of marked formulas).
Let τ and σ be recursive types.

(i) A finite tree of consequences *with marked formulas* (a f.t.o.c. with m.f.) $\mathcal{C}$ results from a f.t.o.c. $\mathcal{C}^{(u)}$ of $\tau = \sigma$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ by attaching superscript-markers to some occurrences of formulas in $\mathcal{C}^{(u)}$ (but we do not require that markers have to be attached: we stipulate $\mathcal{C}^{(u)}$ to be a f.t.o.c. with m.f. itself); the depth of $\mathcal{C}$ is defined to be the depth of $\mathcal{C}^{(u)}$, i.e. $|\mathcal{C}| =_{\text{def}} |\mathcal{C}^{(u)}|$.

(ii) Let a f.t.o.c. with m.f. $\mathcal{C}$ from $\tau = \sigma$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ of the form

$$\begin{array}{c} \tau = \sigma \\ \mathcal{C} \\ (\rho = \chi)^u \end{array}$$

in which one occurrence of the marked formula $(\rho = \chi)^u$ at a leaf-position, for some $\rho, \chi \in \mu Tp$ and an assumption marker u, is fixed.

Then the considered leaf-occurrence of the marked formula $(\rho = \chi)^u$ in $\mathcal{C}$ is called *bound back* iff there exists another occurrence of $(\rho = \chi)^u$ *higher up* in $\mathcal{C}$ (or *inside* $\mathcal{C}$) to which the leaf-occurrence of $(\rho = \chi)^u$ is bound; more precisely, iff there exists a f.t.o.c. with m.f. $\mathcal{C}_1$ from $\tau = \sigma$ in $\mathbf{AK}_\mathbf{0}^{=}$ and a f.t.o.c. with m.f. $\mathcal{C}_2$ from $\rho = \chi$ in $\mathbf{AK}_\mathbf{0}^{=}$ with depth $|\mathcal{C}_2| \geq 1$ such that the derivation-tree $\mathcal{C}$ with the considered leaf-occurrence can also be construed as

$$\begin{array}{c} \tau = \sigma \\ \mathcal{C}_1 \\ (\rho = \chi)^u \\ \mathcal{C}_2 \\ (\rho = \chi)^u \end{array} \tag{6.13}$$

If $\mathcal{C}$ with the considered leaf-occurrence of $(\rho = \chi)^u$ is in fact of the form (6.13), for some f.t.o.c. with m.f. $\mathcal{C}_1$ from $\tau = \sigma$ in $\mathbf{AK}_\mathbf{0}^{=}$ and a f.t.o.c. with m.f. $\mathcal{C}_2$ from $\rho = \chi$ in $\mathbf{AK}_\mathbf{0}^{=}$ such that $|\mathcal{C}_2| \geq 1$, and if it is furthermore the case that in the thread in $\mathcal{C}_2$ from the root-occurrence of $(\rho = \chi)^u$ down to the displayed and considered leaf-occurrence of $(\rho = \chi)^u$ no other occurrence of $(\rho = \chi)^u$ is encountered, then we say: the considered leaf-occurrence of $(\rho = \chi)^u$ in $\mathcal{C}$ is *bound back* to that occurrence of $(\rho = \chi)^u$ inside $\mathcal{C}$ that appears at the position in $\mathcal{C}$ corresponding to the displayed leaf of $\mathcal{C}_1$ in (6.13) (into which the derivation-tree $\mathcal{C}_2$ from $\rho = \chi$ is substituted).

(iii) Let $\tau, \sigma, \rho, \chi \in \mu Tp$, v an assumption marker, and let $\mathcal{C}$ be a f.t.o.c. with m.f. from $\tau = \sigma$ in $\mathbf{AK}_\mathbf{0}^{=}$.

A leaf-occurrence of a marked formula $(\rho = \chi)^v$ in $\mathcal{C}$ that is not bound back is called an *unbound leaf-occurrence* of $(\rho = \chi)^v$) in $\mathcal{C}$.

⊠

Definition 6.3.5 (Finite trees of consequences with back-bound leaf-occurrences of marked formulas).
Let τ and σ be recursive types. A *finite tree of consequences with back-bound leaf-occurrences of marked formulas* (a f.t.o.c. with b.l.o.m.f.) $\mathcal{C}$ from the equation $\tau = \sigma$ in $\mathbf{AK}_\mathbf{0}^{=}$ is a f.t.o.c. with m.f. that satisfies the following two conditions:

(A) Each leaf at the bottom of $\mathcal{C}$ carries either an unmarked formula $\rho = \rho$ for some $\rho \in \mu Tp$ (which corresponds to a reflexivity axioms of equational logic **EQL**) or a marked formula $(\rho = \chi)^u$ for some $\rho, \chi \in \mu Tp$ and some marker u (which leaf-occurrence of $(\rho = \chi)^u$ may or may not be bound back).

(B) For each occurrence of a marked formula $(\rho = \chi)^u$ *inside* $\mathcal{C}$ (for some marker u and some $\rho, \chi \in \mu Tp$), i.e. such that $\mathcal{C}$ with this occurrence displayed can be written as

$$\begin{array}{c} \tau = \sigma \\ \mathcal{C}_1 \\ (\rho = \chi)^u \\ \mathcal{C}_2 \end{array}$$

for some f.t.o.c. with m.f. $\mathcal{C}_1$ from $\tau = \sigma$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ and a f.t.o.c. $\mathcal{C}_2$ from $\rho = \chi$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ with depth $|\mathcal{C}_2| \geq 1$, the following is the case: there exists a leaf-occurrence of $(\rho = \chi)^u$ in $\mathcal{C}$ at a position corresponding to a leaf-occurrence of $(\rho = \chi)^u$ in $\mathcal{C}_2$) which is bound back to the considered occurrence of $(\rho = \chi)^u$ inside $\mathcal{C}$ (the derivation-tree $\mathcal{C}$ can then be written in the form (6.13) with respect to $\mathcal{C}_1$, $\mathcal{C}_2$ and the two occurrences of $(\rho = \chi)^u$, which are linked through the binding).

⊠

Example 6.3.6. It is easy to check that both of the derivation-trees $\mathcal{C}$ of Figure 6.1 and Refl($\mathcal{D}$) of Figure 6.4 are f.t.o.c.'s with b.l.o.m.f. of the equation (6.12) in $\mathbf{AK}_{\mathbf{0}}^{=}$ that have the particular property that all leaf-occurrences of marked formulas in them are bound back. ■

In analogy with notation used in [TS00] for natural-deduction prooftrees with open assumption classes, we extend the symbolic treatment of f.t.o.c.'s described in Notation 6.3.3 to cover also the case of f.t.o.c.'s with b.l.o.m.f. with classes of unbound leaf-occurrences of marked formulas.

Notation 6.3.7 (Symbolic representation of f.t.o.c.'s with b.l.o.m.f.). We will use the notational convention of the below item (i) and the symbolic denotations for f.t.o.c.'s with b.l.o.m.f. in the manner as indicated in item (ii):

(i) By a marked equation $(\tau = \sigma)^{\boldsymbol{m}}$ between recursive types τ and σ and with a boldface-marker $\boldsymbol{m}$ *either* the unmarked equation $\tau = \sigma$ is meant *or* a marked equation of the form $(\tau = \sigma)^{\boldsymbol{m}}$, in which case $\boldsymbol{m}$ is then also assumed to stand for a concrete assumption marker as a syntactical variable varying through assumption markers. But on the other hand, by a marked equation $(\tau = \sigma)^u$ for some non-boldface marker u *only* this particular marked formula is be meant.

(ii) A symbolic derivation-tree

$$\begin{array}{c} (\tau = \sigma)^{\boldsymbol{m}} \\ \mathcal{C} \\ [\rho = \chi]^u \end{array}$$

will be used to denote a f.t.o.c. with b.l.o.m.f. from $\tau = \sigma$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ in which the root carries either the unmarked formula $\tau = \sigma$ or the marked formula $(\tau = \sigma)^{\boldsymbol{m}}$, and where, for some $\rho, \chi \in \mu Tp$ and some marker u, through the symbolic representation of $\mathcal{C}$ the class of all *such* leaf-occurrences of marked formulas $(\rho = \chi)^u$ in $\mathcal{C}$ is considered fixed as the displayed class $[\rho = \chi]^u$ that are unbound in $\mathcal{C}$ (in particular this implies $x \not\equiv \boldsymbol{m}$). However, it will be allowed to consider derivation-trees $\mathcal{C}$ of the form

$$\begin{array}{c} \dfrac{(\tau = \sigma)^u}{(\tau = \sigma)^{\boldsymbol{m}}} \\ \mathcal{C}_1 \\ [\rho = \chi]^u \end{array}$$

where there are leaf-occurrences in $\mathcal{C}_1$ that are unbound in $\mathcal{C}_1$ (indicated by the displayed open assumption class $[\tau = \sigma]^u$ in $\mathcal{C}_1$), but that are bound in $\mathcal{C}$, and in particular, bound back to the root of $\mathcal{C}$.

☒

We continue with an useful lemma that implies that sub-derivation-trees[7] of f.t.o.c.'s with b.l.o.m.f. in $\mathbf{AK}_{\mathbf{0}}^{=}$ are themselves f.t.o.c.'s with b.l.o.m.f. in $\mathbf{AK}_{\mathbf{0}}^{=}$.

Lemma 6.3.8. *Let* $\tau, \sigma, \tau_0, \sigma_0, \tau_1, \sigma_1, \sigma_2, \sigma_2 \in \mu Tp$, *and let* $\mathcal{C}$ *be a f.t.o.c. with b.l.o.m.f. in* $\mathbf{AK}_{\mathbf{0}}^{=}$ *of the form*

$$\begin{array}{c}\dfrac{(\tau = \sigma)^{\boldsymbol{m}}}{(\tau_0 = \sigma_0)^{\boldsymbol{m}_1}}\,R\\ \mathcal{C}_0\end{array} \qquad \textit{or} \qquad \begin{array}{cc} \multicolumn{2}{c}{(\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2)^{\boldsymbol{m}}} \\ \hdashline (\tau_1 = \sigma_1)^{\boldsymbol{m}_1} & (\tau_2 = \sigma_2)^{\boldsymbol{m}_2} \\ \mathcal{C}_1 & \mathcal{C}_2 \end{array}\ \text{DECOMP}$$

(where $R \in \{\text{UNFOLD}_{l/r}, \text{REN}, (\mu{-}\bot)^{\text{der}\bot}_{l/r}\}$*) from the equation* $\tau = \sigma$, *or respectively, from the equation* $\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2$.

Then accordingly $\mathcal{C}_0$, *or both* $\mathcal{C}_1$ *and* $\mathcal{C}_2$ *are f.t.o.c.'s with b.l.o.m.f. in* $\mathbf{AK}_{\mathbf{0}}^{=}$ *from* $\tau_0 = \sigma_0$, *or from* $\tau_1 = \sigma_1$ *and respectively from* $\tau_2 = \sigma_2$.

Proof. This lemma is an easy consequence of the definition of f.t.o.c.'s with b.l.o.m.f. in Definition 6.3.5. □

We proceed with a lemma whose assertion immediately implies that no leaf-occurrence of a marked formula in a f.t.o.c. with b.l.o.m.f. in $\mathbf{AK}_{\mathbf{0}}^{=}$ can be bound back to the premise of a rule $(\mu{-}\bot)^{\text{der}\bot}_{l/r}$. As will also become apparent from the proof of this lemma, the reason for the non-existence of *such* f.t.o.c.'s with b.l.o.m.f. in $\mathbf{AK}_{\mathbf{0}}^{=}$ is similar, in fact 'dual', to the argument given in Remark 6.2.5 for the fact why no rules $(\mu{-}\bot)^{\bot\text{der}}_{l/r}$/FIX have been taken up into the system $\mathbf{e}\text{-}\mathbf{HB}_{\mathbf{0}}^{=}$. There we saw that immediate subderivations of applications of $(\mu{-}\bot)^{\bot\text{der}}_{l/r}$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ cannot be contractive with respect to any class of open marked assumptions. Here we will see that no leaf-occurrence of a marked formula in an f.t.o.c. $\mathcal{C}$ can be bound back across a DECOMP-branching to the premise of a rule $(\mu{-}\bot)^{\text{der}\bot}_{l/r}$.

[7] We have not defined the notion of a "sub-f.t.o.c. in $\mathbf{AK}_{\mathbf{0}}^{=}$", which is referred to informally at this point. However, it would be easy to give a formal definition of this notion along the inductive clauses of Definition 6.3.1. For instance, with respect to the inductive clause (iii) in Definition 6.3.1 we would stipulate that a sub-f.t.o.c. of the f.t.o.c. $\mathcal{C}$, which is defined there from the f.t.o.c.'s $\mathcal{C}_1$ and $\mathcal{C}_2$, to be either $\mathcal{C}$ itself or a sub-f.t.o.c. of $\mathcal{C}_1$ or $\mathcal{C}_2$.

Lemma 6.3.9. *Let $\tau, \sigma, \tilde{\tau}, \tilde{\sigma} \in \mu Tp$ be arbitrary.*

(i) There does not exist a f.t.o.c. with b.l.o.m.f. from $\tau = \sigma$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ of the form

$$\begin{array}{c} \dfrac{(\tau = \sigma)^u}{\tau_1 = \sigma_1}\,(\mu{-}\bot)^{der\bot}_{l/r} \\ \mathcal{C}_1 \end{array} \tag{6.14}$$

where $\tau_1, \sigma_1 \in \mu Tp$ and u an assumption marker.

(ii) What is more, no f.t.o.c. with b.l.o.m.f. of $\tilde{\tau} = \tilde{\sigma}$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ contains an occurrence of a marked formula as the premise of an application of $(\mu{-}\bot)^{der\bot}_{l/r}$.

Proof. Due to Lemma 6.3.8, part (ii) of the lemma is an easy consequence of part (i). Thus it suffices to show item (i) and hence that there cannot be a f.t.o.c. with b.l.o.m.f. in $\mathbf{AK}_{\mathbf{0}}^{=}$ whose root at the top is a marked formula that is immediately succeeded by an application of a rule $(\mu{-}\bot)^{\mathrm{der}\bot}_{l/r}$.

To show this, let $\tau, \sigma \in \mu Tp$ be arbitrary, and we *assume* that there exists a f.t.o.c. with b.l.o.m.f. $\mathcal{C}$ of $\tau = \sigma$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ of the form (6.14) in which the formula at the root of is the upper premise of an application of $(\mu{-}\bot)^{\mathrm{der}\bot}_{l}$ (for the case with an application of $(\mu{-}\bot)^{\mathrm{der}\bot}_{r/l}$ it can be argued analogously). We will show a contradiction from this assumption with the fact that $\mathcal{C}$ is a f.t.o.c. with b.l.o.m.f. in $\mathbf{AK}_{\mathbf{0}}^{=}$. By our assumption, $\mathcal{C}$ is of the form

$$\begin{array}{c} \dfrac{(\tau = \sigma)^u}{\bot = \sigma}\,(\mu{-}\bot)^{\mathrm{der}\bot}_{l} \\ \mathcal{C}_1 \\ [\tau = \sigma]^u \end{array}$$

with $\tau \equiv \mu\alpha\alpha_1 \ldots \alpha_n . \alpha$, for some $n \in \omega$ and $\alpha, \alpha_1, \ldots, \alpha_n \in TVar$, and in particular with $\tau \not\equiv \bot$. Since $\mathcal{C}$ is a f.t.o.c. with b.l.o.m.f. in $\mathbf{AK}_{\mathbf{0}}^{=}$, there exists at least one leaf-occurrence of $(\tau = \sigma)^u$ in $\mathcal{C}_1$ that is bound back in $\mathcal{C}$ to the formula at the top (the class $[\tau = \sigma]^u$ displayed at the bottom of $\mathcal{C}_1$, and hence of $\mathcal{C}$, is intended to symbolize the inhabited class of all those leaf-occurrences of $(\tau = \sigma)^u$ in $\mathcal{C}$ that are bound back to the root of $\mathcal{C}$). However, the immediate subtree $\mathcal{C}_1$ of $\mathcal{C}$ can only be of the form

$$\frac{\bot = \sigma}{\bot = \sigma_1}\,(\mathrm{REN},\ \mathrm{FOLD}_r,\ (\mu{-}\bot)^{\mathrm{der}\bot}_{r})^*$$

for some $\sigma_1 \in \mu Tp$, containing only applications of the rules REN, FOLD_r and $(\mu{-}\bot)^{\mathrm{der}\bot}_{r}$ since there is no rule application in $\mathbf{AK}_{\mathbf{0}}^{=}$, let alone a branching that is able change $\bot$ as the recursive type on the left side of an arbitrary equation in $\mathcal{C}_1$. Thus $\mathcal{C}$ cannot contain a single branching DECOMP and since $\tau \not\equiv \bot$, it is furthermore the case that $\mathcal{C}_1$ and $\mathcal{C}$ do not contain any leaf occurrence of $(\tau = \sigma)^u$. Therefore both conditions (I) and (II) in Definition 6.3.5 are violated, and hence, $\mathcal{C}$ is not a f.t.o.c. with b.l.o.m.f., in contradiction with our assumption.

In this way we also have also item (i) of the lemma.

□

For the purpose of being able to define inductively a reflection function $\boldsymbol{\mathcal{D}}$ for f.t.o.c.'s with b.l.o.m.f. in $\mathbf{AK}_{\mathbf{0}}^{=}$, it will be useful to have also an alternative and inductive direct definition of f.t.o.c.'s with b.l.o.m.f.'s at hand. We have chosen not to give a second name to this notion, but use the term "f.t.o.c. with b.l.o.m.f." again in the following Definition and Lemma that also states the equivalence of the two definitions.

Definition and Lemma 6.3.10 (Alternative inductive definition of f.t.o.c.'s with b.l.o.m.f.). The notion defined in Definition 6.3.5 of "finite tree of consequences with back-bound leaf-occurrences of marked formulas" can be defined equivalently as follows.

A *f.t.o.c. with b.l.o.m.f.* $\mathcal{C}$ of $\tau = \sigma$ in $\mathbf{AK}_{\mathbf{0}}^{=}$, where $\tau, \sigma \in \mu Tp$, is an object that together with the assertion "$\mathcal{C}$ is a ..." can be formed by a finite number of applications of the six generation rules (i)–(vi) given below. Thereby simultaneously the *depth* of $\mathcal{C}$ is defined as well as the notion of such *leaf-occurrences of marked formulas* (*l.o. of m.f.*) in $\mathcal{C}$ that are *not bound back*, or *unbound*. For the purpose of an illustrative exhibition of the generation rules, we have put the symbolic f.t.o.c.'s with b.l.o.m.f. into framed boxes that are *not* part of the defined objects.

(i) For all $\tau, \sigma \in \mu Tp$ and assumption markers u, $\boxed{(\tau = \sigma)^u}$ is a f.t.o.c. with b.l.o.m.f. $\mathcal{C}$ from $\tau = \sigma$ in $\mathbf{AK}_{\mathbf{0}}^{=}$. Here $|\mathcal{C}| =_{\text{def}} 0$ and the only unbound l.o. of a m.f. in $\mathcal{C}$ is the single occurrence of $(\tau = \sigma)^u$ in $\mathcal{C}$ itself.

(ii) For all $\tau \in \mu Tp$, $\boxed{\tau = \tau}$ is a f.t.o.c. with b.l.o.m.f. $\mathcal{C}$ from $\tau = \tau$ in $\mathbf{AK}_{\mathbf{0}}^{=}$. Here $|\mathcal{C}| =_{\text{def}} 0$ and $\mathcal{C}$ does not contain any unbound leaf-occurrences of marked formulas.

(iii) For all $\tau, \sigma, \tau_1, \sigma_1 \in \mu Tp$, $\boxed{\begin{array}{c} \dfrac{\tau = \sigma}{(\tau_1 = \sigma_1)^{\boldsymbol{m}_1}}\, R \\ \mathcal{C}_1 \end{array}}$ is a f.t.o.c. with b.l.o.m.f. $\mathcal{C}$ from $\tau = \sigma$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ given that $\mathcal{C}_1$ is a f.t.o.c. with b.l.o.m.f. from $\tau_1 = \sigma_1$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ and that R is an application of a rule UNFOLD$_{l/r}$, REN or $(\mu\text{–}\bot)^{\text{der}\bot}_{l/r}$ in $\mathbf{AK}_{\mathbf{0}}^{=}$.

Here we set $|\mathcal{C}| =_{\text{def}} |\mathcal{C}_1| + 1$ and stipulate that the unbound l.o. of m.f. in $\mathcal{C}$ correspond uniquely and in an obvious way to respective unbound l.o. of m.f. in its part $\mathcal{C}_1$.

(iv) For all $\tau, \sigma, \tau_1, \sigma_1 \in \mu Tp$ and all assumption markers u, $\boxed{\begin{array}{c} \dfrac{(\tau = \sigma)^u}{(\tau_1 = \sigma_1)^{\boldsymbol{m}_1}}\, R \\ \mathcal{C}_1 \\ [\tau = \sigma]^u \end{array}}$

is a f.t.o.c. with b.l.o.m.f. $\mathcal{C}$ of $\tau = \sigma$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ given that $\mathcal{C}_1$ is a f.t.o.c. with b.l.o.m.f. of $\tau_1 = \sigma_1$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ that contains at least one unbound l.o. of $(\tau = \sigma)^u$, and given that R is an application of a rule UNFOLD$_{l/r}$ or REN in $\mathbf{AK}_{\mathbf{0}}^{=}$. *Note* that in this case rules $(\mu - \bot)^{\mathrm{der}\bot}_{l/r}$ are not allowed as rule applications at the top of $\mathcal{C}$.

Here we have again $|\mathcal{C}| =_{\mathrm{def}} |\mathcal{C}_1| + 1$. All unbound l.o. of the marked formula $(\tau = \sigma)^u$ in $\mathcal{C}_1$ correspond uniquely to such l.o. of this m.f. in $\mathcal{C}$ that are bound back to the occurrence of $(\tau = \sigma)^u$ at the root of $\mathcal{C}$; hence $\mathcal{C}$ does not contain unbound l.o. of $(\tau = \sigma)^u$. For all marked formulas $(\rho = \chi)^v$ different from $(\tau = \sigma)^u$ it holds that all unbound l.o. of $(\rho = \chi)^v$ in $\mathcal{C}$ correspond uniquely and in an obvious way to unbound l.o. of $(\rho = \chi)^v$ in its part $\mathcal{C}_1$.

(v) For all $\tau_1, \tau_2, \sigma_1, \sigma_2 \in \mu Tp$,

$$\boxed{\begin{array}{c} \tau_1 \to \tau_2 = \sigma_1 \to \sigma_2 \\ \hdashline (\tau_1 = \sigma_1)^{\boldsymbol{m_1}} \qquad (\tau_2 = \sigma_2)^{\boldsymbol{m_2}} \\ \mathcal{C}_1 \qquad\qquad\qquad \mathcal{C}_2 \end{array}\ \text{DECOMP}}$$

is a f.t.o.c. with b.l.o.m.f. $\mathcal{C}$ from $\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ given that, for each $i \in \{1, 2\}$, $\mathcal{C}_i$ is a f.t.o.c. with b.l.o.m.f. from $\tau_i = \sigma_i$ in $\mathbf{AK}_{\mathbf{0}}^{=}$.

Here we set $|\mathcal{C}| =_{\mathrm{def}} 1 + \max\{|\mathcal{C}_1|, |\mathcal{C}_2|\}$, and we stipulate that the unbound l.o. of m.f. in $\mathcal{C}$ correspond uniquely and in an obvious way to the the unbound l.o. of m.f. in either of its parts $\mathcal{C}_1$ or $\mathcal{C}_2$.

(vi) For all $\tau_1, \tau_2, \sigma_1, \sigma_2 \in \mu Tp$,

$$\boxed{\begin{array}{c} (\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2)^u \\ \hdashline (\tau_1 = \sigma_1)^{\boldsymbol{m_1}} \qquad (\tau_2 = \sigma_2)^{\boldsymbol{m_2}} \\ \mathcal{C}_1 \qquad\qquad\qquad \mathcal{C}_2 \\ [\tau = \sigma]^u \qquad\qquad [\tau = \sigma]^u \end{array}\ \text{DECOMP}}$$

is a f.t.o.c. with b.l.o.m.f. $\mathcal{C}$ from $\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ given that, for each $i \in \{1, 2\}$, $\mathcal{C}_i$ is a f.t.o.c. with b.l.o.m.f. from $\tau_i = \sigma_i$ in $\mathbf{AK}_{\mathbf{0}}^{=}$, and that there is at least one unbound leaf-occurrence of the marked formula $(\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2)^u$ in either $\mathcal{C}_1$ or in $\mathcal{C}_2$.

Here we set again $|\mathcal{C}| =_{\mathrm{def}} 1 + \max\{|\mathcal{C}_1|, |\mathcal{C}_2|\}$. All unbound l.o. of the marked formula $(\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2)^u$ in $\mathcal{C}_1$ or in $\mathcal{C}_2$ correspond uniquely to l.o. of this m.f. in $\mathcal{C}$ that are bound back to the occurrence of this formula at the root of $\mathcal{C}$; hence there do not exist unbound leaf-occurrence of this marked formula in $\mathcal{C}$. For all m.f.'s $(\rho = \chi)^v$ different from $(\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2)^u$, it holds that unbound l.o. of $(\rho = \chi)^v$ in $\mathcal{C}$ correspond uniquely and in an obvious way to unbound l.o. of $(\rho = \chi)^v$ in either of its parts $\mathcal{C}_1$ or in $\mathcal{C}_2$.

Proof. Both directions in the equivalence of the previous definition in Definition 6.3.5 with the alternative one here can be done by easy inductions on the depth $|\mathcal{C}|$ of respectively defined f.t.o.c. with b.l.o.m.f. in $\mathbf{AK}_{\mathbf{0}}^{=}$. In the induction for showing the slightly less immediate direction, that a f.t.o.c. with b.l.o.m.f. $\mathcal{C}$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ with respect to Definition 6.3.5 is also a f.t.o.c. with b.l.o.m.f. according to the new definition, both Lemma 6.3.8 and Lemma 6.3.9 are essential. □

By relying on the notion of f.t.o.c. with b.l.o.m.f. in $\mathbf{AK}_{\mathbf{0}}^{=}$, we are now able to define "consistency-unfoldings in $\mathbf{AK}_{\mathbf{0}}^{=}$", which are formalizations of derivation-trees that correspond to successful consistency-checks in $\mathbf{AK}_{\mathbf{0}}^{=}$ as considered in Section 6.1. For this, we introduce the auxiliary notion of "partial consistency-unfolding in $\mathbf{AK}_{\mathbf{0}}^{=}$" first.

Definition 6.3.11 ((Partial) Consistency-Unfoldings). Let $\tau, \sigma \in \mu Tp$.

(i) A *partial consistency-unfolding* $\mathcal{C}$ of the equation $\tau = \sigma$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ is a f.t.o.c. $\mathcal{C}$ with b.l.o.m.f. from $\tau = \sigma$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ that satisfies the following property: for all leaf-occurrences of marked formulas in $\mathcal{C}$ that are bound back there exists at least one occurrence of a branching DECOMP in the thread in $\mathcal{C}$ up to that occurrence of a marked formula to which the leaf-occurrence is bound back.

(ii) A *consistency-unfolding* $\mathcal{C}$ of the equation $\tau = \sigma$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ is a partial consistency-unfolding $\mathcal{C}$ of $\tau = \sigma$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ in which each leaf-occurrence of a marked formula is bound back.

By a *partial consistency-unfolding in* $\mathbf{AK}_{\mathbf{0}}^{=}$ (a *consistency-unfolding in* $\mathbf{AK}_{\mathbf{0}}^{=}$) we mean a partial consistency-unfolding (a consistency-unfolding) in $\mathbf{AK}_{\mathbf{0}}^{=}$ of the equation $\tilde{\tau} = \tilde{\sigma}$ for some $\tilde{\tau}, \tilde{\sigma} \in \mu Tp$. And furthermore, we denote by $pCU(\mathbf{AK}_{\mathbf{0}}^{=})$, and by $CU(\mathbf{AK}_{\mathbf{0}}^{=})$, the set of partial consistency-unfoldings in $\mathbf{AK}_{\mathbf{0}}^{=}$, and respectively, the set of consistency-unfoldings in $\mathbf{AK}_{\mathbf{0}}^{=}$.

☒

Example 6.3.12. Since we have already recognized both of our running examples, the derivation-trees $\mathcal{C}$ of Figure 6.1 and Refl($\mathcal{D}$) of Figure 6.4, as f.t.o.c.'s with b.l.o.m.f. from (6.12) in $\mathbf{AK}_{\mathbf{0}}^{=}$, it is easy to verify that both of them are in fact consistency-unfoldings in $\mathbf{AK}_{\mathbf{0}}^{=}$ of (6.12). ■

Having formally introduced consistency-unfoldings in $\mathbf{AK}_{\mathbf{0}}^{=}$ and proposed them to be adequate formalizations of successful consistency-checks with respect to $\mathbf{AK}_{\mathbf{0}}^{=}$ in the way of Example 6.1.1, we still have to justify that consistency-unfoldings in $\mathbf{AK}_{\mathbf{0}}^{=}$ indeed 'witness' the consistency with respect to $\mathbf{AK}_{\mathbf{0}}^{=}$ of the equation at their root, in the sense that they enable straightforward inductive proofs for showing this. That is, we have to prove the following theorem.

Theorem 6.3.13. *For all recursive types $\tau, \sigma \in \mu Tp$ it holds that:*

$$(\exists \mathcal{C}) \begin{bmatrix} \mathcal{C} \textit{ is a consistency-unfol-} \\ \textit{ding of } \tau = \sigma \textit{ in } \mathbf{AK}_{\mathbf{0}}^{=} \end{bmatrix} \quad \Longrightarrow \quad \tau = \sigma \textit{ is } \mathbf{AK}_{\mathbf{0}}^{=}\textit{-consistent}\,. \tag{6.15}$$

We will prove this theorem below on page 176 after having stated and proved four necessary lemmas. The first and the second one below assert that leading symbols of recursive types are respectively invariant under the conversion $\twoheadleftarrow\!\!\twoheadrightarrow_{\mathrm{r/o\text{-}u}(\mu\perp)'}$ that belongs to the reduction relation $\to_{\text{r/o-u}(\mu\bot)'}$ introduced in Definition 5.3.5, and under $\mathbf{AK}_{\mathbf{0}}^{=}$-consequences of equations between $\to_{\text{r/o-u}(\mu\bot)'}$-convertible recursive types.

Lemma 6.3.14. *For all $\tau, \sigma \in \mu Tp$ it holds:*

$$\tau \twoheadleftarrow\!\!\twoheadrightarrow_{r/o\text{-}u(\mu\bot)'} \sigma \quad\Longrightarrow\quad \mathcal{L}'(\tau) = \mathcal{L}'(\sigma)\ . \tag{6.16}$$

Proof. The reduction relations $\to_{\text{ren}}$, $\to_{\text{out-unf}}$ and $\to_{\text{r/o-u}(\mu\bot)'}$ have the property that they do not change the tree unfolding, and hence neither the leading symbol, of a recursive type to which they are applied. Therefore the assertion of the lemma follows, for all $\tau, \sigma \in \mu Tp$, by induction on the length of a conversion $\tau \twoheadleftarrow\!\!\twoheadrightarrow_{\text{r/o-u}(\mu\bot)'} \sigma$ between τ and σ. □

Lemma 6.3.15. *For all $\tau, \sigma, \chi_1, \chi_2 \in \mu Tp$ it holds:*

$$(\,\tau \twoheadleftarrow\!\!\twoheadrightarrow_{r/o\text{-}u(\mu\bot)'} \sigma\,)\ \&\ (\,\tau = \sigma \vdash_{\mathbf{AK}_{\mathbf{0}}^{=}} \chi_1 = \chi_2\,) \quad\Longrightarrow\quad \mathcal{L}'(\chi_1) = \mathcal{L}'(\chi_2)\ . \tag{6.17}$$

Proof. The assertion of the lemma can be shown by proving (6.17), for all $\tau, \sigma \in \mu Tp$, with induction on the depth $|\mathcal{D}|$ of derivations $\mathcal{D}$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ from assumption $\tau = \sigma$ and with conclusion $\chi_1 = \chi_2$ (with some $\chi_1, \chi_2 \in \mu Tp$). Hereby the base case of the induction follows from Lemma 6.3.14. In the induction step a case-distinction is made according to which rule R of $\mathbf{AK}_{\mathbf{0}}^{=}$ is applied first in $\mathcal{D}$, immediately below the assumption $\tau = \sigma$. If R is one of the rules $\text{UNFOLD}_{l/r}$, $(\mu{-}\bot)^{\text{der}\bot}_{l/r}$ or REN, then $\mathcal{D}$ is of the form

$$\begin{array}{c} \dfrac{\overbrace{\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2}^{\tau = \sigma}}{(\,\tau_i = \sigma_i\,)}\ \text{DECOMP} \\ \mathcal{D}_1 \\ \chi_1 = \chi_2 \end{array}$$

for some $\tau_1, \tau_2, \sigma_1, \sigma_2 \in \mu Tp$ and $i \in \{1, 2\}$ such that $\tau \equiv \tau_1 \to \tau_2$ and $\sigma \equiv \sigma_1 \to \sigma_2$, and whereby assumption $\tau_1 \to \tau_2 \twoheadleftarrow\!\!\twoheadrightarrow_{\text{ren/out-unf}} \sigma_1 \to \sigma_2$ holds. Here Lemma 5.3.6, (iv), implies $\tau_i \equiv_{\text{ren}} \sigma_i$, and hence $\tau_i \twoheadleftarrow\!\!\twoheadrightarrow_{\text{r/o-u}(\mu\bot)'} \sigma_i$, from which an application of the induction hypothesis to the subderivation $\mathcal{D}_1$ of $\mathcal{D}$ gives $\mathcal{L}'(\chi_1) = \mathcal{L}'(\chi_2)$. □

The third lemma asserts that consistency-unfoldings in $\mathbf{AK}_{\mathbf{0}}^{=}$ do not contain contradictions with respect to $=_\mu$.

Lemma 6.3.16. *Let $\mathcal{C}$ be a consistency-unfolding in $\mathbf{AK}_{\mathbf{0}}^{=}$, and let $\chi_1 = \chi_2$ be an equation between recursive types that occurs in $\mathcal{C}$. Then $\mathcal{L}'(\chi_1) = \mathcal{L}'(\chi_2)$ holds, i.e. the recursive types χ_1 and χ_2 have the same leading symbol.*

Proof. Let $\mathcal{C}$ be a consistency-unfolding in $\mathbf{AK}_{\mathbf{0}}^{=}$. For the argument below, let us fix an arbitrary equation $\chi_1 = \chi_2$ in $\mathcal{C}$.

For showing that the recursive types χ_1 and χ_2 in the equation of which we have fixed an occurrence in $\mathcal{C}$ have the same leading symbol, we distinguish three cases about the position of this considered occurrence within $\mathcal{C}$.

Case 1. There is an occurrence of a DECOMP-branching in $\mathcal{C}$ below the considered occurrence of $\chi_1 = \chi_2$.

Then by following an arbitrary thread in $\mathcal{C}$ from the considered occurrence of $\chi_1 = \chi_2$ downwards to the upper premise $\rho_{11} \to \rho_{12} = \rho_{21} \to \rho_{22}$ of the first encountered DECOMP-branching, a derivation

$$\frac{\chi_1 = \chi_2}{\rho_{11} \to \rho_{12} = \rho_{21} \to \rho_{22}} \;(\text{UNFOLD}_{l/r},\, \text{REN})^*$$

is found that consists only of applications of rules $\text{UNFOLD}_{l/r}$ and REN. Hence

$$\chi_1 \twoheadrightarrow_{\text{r/o-u}(\mu\perp)'} \rho_{11} \to \rho_{12} \qquad \text{and} \qquad \chi_2 \twoheadrightarrow_{\text{r/o-u}(\mu\perp)'} \rho_{21} \to \rho_{22}$$

follows, which because of $\mathcal{L}'(\rho_{11} \to \rho_{12}) = \to = \mathcal{L}'(\rho_{21} \to \rho_{22})$ entails

$$\mathcal{L}'(\chi_1) = \mathcal{L}'(\chi_2) \tag{6.18}$$

by Lemma 6.3.14.

Case 2. There does not occur a DECOMP-branching in $\mathcal{C}$ below the considered occurrence of $\chi_1 = \chi_2$, and the single leaf of $\mathcal{C}$ below this considered occurrence is one that is carrying the formula $\rho = \rho$ for some $\rho \in \mu Tp$.

Then by following the unique thread in $\mathcal{C}$ from the considered occurrence of $\chi_1 = \chi_2$ downwards to the leaf-occurrence of $\rho = \rho$, an $\mathbf{AK}_{\mathbf{0}}^{=}$-derivation of the form

$$\frac{\chi_1 = \chi_2}{\rho = \rho} \;(\text{UNFOLD}_{l/r},\, \text{REN},\, (\mu{-}\perp)^{\text{der}\perp}_{l/r})^*$$

can be extracted that consists only of applications of the rules $\text{UNFOLD}_{l/r}$, REN, and $(\mu{-}\perp)^{\text{der}\perp}_{l/r}$. Here we find

$$\chi_1 \twoheadrightarrow_{\text{r/o-u}(\mu\perp)'} \rho \qquad \text{and} \qquad \chi_2 \twoheadrightarrow_{\text{r/o-u}(\mu\perp)'} \rho$$

which, again due to Lemma 6.3.14, implies (6.18).

Case 3. There does not occur a DECOMP-branching in $\mathcal{C}$ below the considered occurrence of $\chi_1 = \chi_2$, and at the single leaf of $\mathcal{C}$ below this considered occurrence a marked formula $(\rho_1 = \rho_2)^u$ occurs, for some $\rho_1, \rho_2 \in \mu Tp$ and an assumption marker u, that is bound back in $\mathcal{C}$.

Since $\mathcal{C}$ is a consistency-unfolding in $\mathbf{AK}_{\mathbf{0}}^{=}$, there must occur at least one branching DECOMP below that occurrence of $(\rho_1 = \rho_2)^u$ in $\mathcal{C}$ to which the leaf-occurrence of $(\rho_1 = \rho_2)^u$ below the considered occurrence of $\chi_1 = \chi_2$ is bound back. Hence by first following the thread in $\mathcal{C}$ from the considered occurrence of $\chi_1 = \chi_2$ down to the leaf-occurrence of $(\rho_1 = \rho_2)^u$, and by then descending from that occurrence of $(\rho_1 = \rho_2)^u$ to the nearest branching DECOMP, which we assume has the premise $\rho_{11} \to \rho_{12} = \rho_{21} \to \rho_{22}$, a derivation in $\mathbf{AK}_{\mathbf{0}}^{=}$ of the form

$$\dfrac{\dfrac{\chi_1 = \chi_2}{\rho_1 = \rho_2}\,(\text{UNFOLD}_{l/r},\, \text{REN})^*}{\rho_{11} \to \rho_{12} = \rho_{21} \to \rho_{22}}\,(\text{UNFOLD}_{l/r},\, \text{REN})^*$$

can be extracted that contains only occurrences of rules $\text{UNFOLD}_{l/r}$ and REN. With this derivation it can now be argued analogously as in Case 1 to show that (6.18) holds here as well.

Due to the fact that consistency-unfoldings do not contain u.l.o.m.f.'s, the cases 1–3 exhaust all possibilities for the position of the considered occurrence of $\chi_1 = \chi_2$ in $\mathcal{C}$. Since we have proved (6.18) in all three cases, we have proved that the recursive types on the left- and on the right-hand side of the considered equation $\chi_1 = \chi_2$ in $\mathcal{C}$ have the same leading symbol.

□

The fourth lemma given below is going to be our main tool for showing Theorem 6.3.13. It asserts a somewhat technical statement about how, for all $\tau, \sigma \in \mu Tp$, an arbitrary derivation in $\mathbf{AK}_{\mathbf{0}}^{=}$ from assumption $\tau = \sigma$ is related to an arbitrary partial consistency-unfolding of $\tau = \sigma$ in $\mathbf{AK}_{\mathbf{0}}^{=}$.

Lemma 6.3.17. *Let $\tau, \sigma, \chi_1, \chi_2 \in \mu Tp$ and let $\mathcal{C}$ be a partial consistency-unfolding in $\mathbf{AK}_{\mathbf{0}}^{=}$. Then for all derivations $\mathcal{D}$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ from the assumption $\tau = \sigma$ and with conclusion $\chi_1 = \chi_2$ one of the following three assertions holds:*

(i) For some $\tilde{\chi}_1, \tilde{\chi}_2 \in \mu Tp$, there is an occurrence of the equation $\tilde{\chi}_1 = \tilde{\chi}_2$ in $\mathcal{C}$ such that it holds:

$$\chi_1 \twoheadleftarrow\!\!\twoheadrightarrow_{r/o\text{-}u(\mu\bot)'} \tilde{\chi}_1 \quad \textit{and} \quad \chi_2 \twoheadleftarrow\!\!\twoheadrightarrow_{r/o\text{-}u(\mu\bot)'} \tilde{\chi}_2 \;.$$

(ii) For some $\rho_1, \rho_2 \in \mu Tp$ and an assumption marker u, there is an unbound leaf-occurrence of a marked formula $(\rho_1 = \rho_2)^u$ in $\mathcal{C}$ such that for some $\tilde{\rho}_1, \tilde{\rho}_2 \in \mu Tp$ it holds:

$$(\,\tilde{\rho}_1 \twoheadleftarrow\!\!\twoheadrightarrow_{r/o\text{-}u(\mu\bot)'} \rho_1\,) \;\;\&\;\; (\,\tilde{\rho}_2 \twoheadleftarrow\!\!\twoheadrightarrow_{r/o\text{-}u(\mu\bot)'} \rho_2\,) \;\;\&\;\; (\,\tilde{\rho}_1 = \tilde{\rho}_2 \vdash_{\mathbf{AK}_{\mathbf{0}}^{=}} \chi_1 = \chi_2\,)\;.$$

(iii) For some $\rho \in \mu Tp$, there is a leaf-occurrence of a equation $\rho = \rho$ in $\mathcal{C}$ such for some $\tilde{\rho}_1, \tilde{\rho}_2 \in \mu Tp$ it holds:

$$(\,\tilde{\rho}_1 \twoheadleftarrow\!\!\twoheadrightarrow_{r/o\text{-}u(\mu\bot)'} \rho\,) \;\;\&\;\; (\,\tilde{\rho}_2 \twoheadleftarrow\!\!\twoheadrightarrow_{r/o\text{-}u(\mu\bot)'} \rho\,) \;\;\&\;\; (\,\tilde{\rho}_1 = \tilde{\rho}_2 \vdash_{\mathbf{AK}_{\mathbf{0}}^{=}} \chi_1 = \chi_2\,)\;.$$

Hint on the proof. This lemma can be shown by induction on the depth $|\mathcal{D}|$ of a derivation $\mathcal{D}$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ from the assumption $\tau = \sigma$ with conclusion $\chi_1 = \chi_2$. The base case is trivial. In the induction step only the case of derivations with an application of DECOMP at the bottom is non-trivial. In this case, however, the use of the cyclic structure of partial consistency-unfoldings with respect to back-binding, and the assertion of Lemma 5.3.6, (iv), are essential. □

We are finally able to carry out a proof for Theorem 6.3.13.

Proof of Theorem 6.3.13. Let $\tau, \sigma \in \mu Tp$ be arbitrary, and let $\mathcal{C}$ be an arbitrary consistency-unfolding in $\mathbf{AK}_{\mathbf{0}}^{=}$. We will show that $\tau = \sigma$ is $\mathbf{AK}_{\mathbf{0}}^{=}$-consistent, i.e. that no derivation in $\mathbf{AK}_{\mathbf{0}}^{=}$ from the assumption $\tau = \sigma$ has a contradiction with respect to $=_\mu$ as its conclusion.

For this, we let $\mathcal{D}$ be an arbitrary derivation in $\mathbf{AK}_{\mathbf{0}}^{=}$ from assumption $\tau = \sigma$ and with conclusion $\chi_1 = \chi_2$, for some $\chi_1, \chi_2 \in \mu Tp$. By Lemma 6.3.17 it follows, since, as a consistency-unfolding, $\mathcal{C}$ does not contain u.l.o.m.f.'s and hence assertion (ii) of Lemma 6.3.17 cannot occur here, that one of the following two assertions

$$
\begin{aligned}
&(\exists\, \tilde{\chi}_1, \tilde{\chi}_2 \in \mu Tp)\, (\exists\, \text{occurrence of } \tilde{\chi}_1 = \tilde{\chi}_2 \text{ in } \mathcal{C}) \\
&\qquad\qquad \big[\, \chi_1 \mathrel{\twoheadleftarrow\!\!\twoheadrightarrow}_{\mathrm{r/o\text{-}u}(\mu\perp)'} \tilde{\chi}_1 \;\;\&\;\; \chi_2 \mathrel{\twoheadleftarrow\!\!\twoheadrightarrow}_{\mathrm{r/o\text{-}u}(\mu\perp)'} \tilde{\chi}_2 \,\big]\,, \qquad (6.19)
\end{aligned}
$$

or

$$
\begin{aligned}
&(\exists \rho, \tilde{\rho}_1, \tilde{\rho}_2 \in \mu Tp)\, \big[\, (\, \tilde{\rho}_1 \mathrel{\twoheadleftarrow\!\!\twoheadrightarrow}_{\mathrm{r/o\text{-}u}(\mu\perp)'} \rho\,) \;\;\& \\
&\qquad \&\;\; (\, \tilde{\rho}_2 \mathrel{\twoheadleftarrow\!\!\twoheadrightarrow}_{\mathrm{r/o\text{-}u}(\mu\perp)'} \rho\,) \;\;\&\;\; (\, \tilde{\rho}_1 = \tilde{\rho}_2 \vdash_{\mathbf{AK}_{\mathbf{0}}^{=}} \chi_1 = \chi_2\,)\,\big]\,. \qquad (6.20)
\end{aligned}
$$

holds. If, on the one hand, (6.19) holds, then it follows by Lemma 6.3.14 that also

$$
\begin{aligned}
&(\exists\, \tilde{\chi}_1, \tilde{\chi}_2 \in \mathcal{C})\, (\exists\, \text{occurrence of } \tilde{\chi}_1 = \tilde{\chi}_2 \text{ in } \mathcal{C}) \\
&\qquad\qquad \big[\, \mathcal{L}'(\chi_1) = \mathcal{L}'(\tilde{\chi}_1) \;\;\&\;\; \mathcal{L}'(\chi_2) = \mathcal{L}'(\tilde{\chi}_2)\,\big] \qquad (6.21)
\end{aligned}
$$

holds. We know from Lemma 6.3.16 that consistency-unfoldings do not contain contradictions with respect to $=_\mu$. Hence, because $\mathcal{C}$ is a consistency-unfolding, (6.21) entails

$$\mathcal{L}'(\chi_1) = \mathcal{L}'(\chi_2)\,. \qquad (6.22)$$

If, on the other hand, (6.20) is the case,

$$(\exists \tilde{\rho}_1, \tilde{\rho}_2 \in \mu Tp)\, \big[\, (\, \tilde{\rho}_1 \mathrel{\twoheadleftarrow\!\!\twoheadrightarrow}_{\mathrm{r/o\text{-}u}(\mu\perp)'} \tilde{\rho}_2\,) \;\;\&\;\; (\, \tilde{\rho}_1 = \tilde{\rho}_2 \vdash_{\mathbf{AK}_{\mathbf{0}}^{=}} \chi_1 = \chi_2\,)\,\big]$$

follows, from which (6.22) is entailed by an application of Lemma 6.3.15. In both cases we have shown (6.22) and hence that the conclusion of $\mathcal{D}$ is not a contradiction with respect to $=_\mu$. Since we have considered an arbitrary $\mathbf{AK}_{\mathbf{0}}^{=}$-derivation $\mathcal{D}$ from $\tau = \sigma$, we have eventually proven that $\tau = \sigma$ is consistent with respect to $\mathbf{AK}_{\mathbf{0}}^{=}$.

□

Theorem 6.3.13 states that consistency-unfoldings in $\mathbf{AK}_{\mathbf{0}}^{=}$ of an equation $\tau = \sigma$ guarantee the consistency of $\tau = \sigma$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ with respect to $=_\mu$. Hence naturally also the question arises whether the concept of "consistency-unfolding" is indeed general enough to capture the notion of consistency of formulas with respect to $\mathbf{AK}_{\mathbf{0}}^{=}$. As stated by the following theorem, the answer is positive. This theorem, for which we are only going to sketch a proof, establishes a link between the notions of "$\mathbf{AK}_{\mathbf{0}}^{=}$-consistency" and "consistency-unfolding in $\mathbf{AK}_{\mathbf{0}}^{=}$".

Theorem 6.3.18. *For all recursive types $\tau, \sigma \in \mu Tp$ it holds that:*

$$(\exists \mathcal{C}) \begin{bmatrix} \mathcal{C} \textit{ is a consistency-unfol-} \\ \textit{ding of } \tau = \sigma \textit{ in } \mathbf{AK}_\mathbf{0}^= \end{bmatrix} \iff \tau = \sigma \textit{ is } \mathbf{AK}_\mathbf{0}^=\textit{-consistent}. \qquad (6.23)$$

Sketch of the Proof. Let $\tau, \sigma \in \mu Tp$. The implication "$\Rightarrow$" in (6.23) is an instance of the implication (6.15) of Theorem 6.3.13.

The implication "$\Leftarrow$" in (6.23) can be shown by an analogous, in fact as good as 'dual', argument to that one used in a proof (following [BrHe98]) for the completeness of $\mathbf{HB}_\mathbf{0}^=$ with respect to $=_\mu$, namely as follows. For an arbitrary given equation $\tau = \sigma$ between recursive types $\tau, \sigma \in \mu Tp$ for which $\tau =_\mu \sigma$ holds, a consistency-unfolding of $\tau = \sigma$ in $\mathbf{AK}_\mathbf{0}^=$ can be reached by building up the "tree of consequences" of this equation in $\mathbf{AK}_\mathbf{0}^=$ in successive extension stages, cutting off branches always as soon as "looping" occurs or as soon as a formula $\chi = \chi$ has been encountered. There cannot be infinite branches in the arising derivation-tree due to the fact that the set of conclusions of derivations from $\tau = \sigma$ in $\mathbf{AK}_\mathbf{0}^=$ is always finite, if equations that arise from each other by applications of REN are not counted separately.

□

In view of Theorem 5.2.13, the correspondence theorem between $\mathbf{AK}^=$-consistency and recursive type equality, the above theorem immediately entails the corollary below.

Corollary 6.3.19. *For all recursive types $\tau, \sigma \in \mu Tp$ it holds that:*

$$(\exists \mathcal{C}) \left[\mathcal{C} \textit{ is a consistency-unfolding of } \tau = \sigma \textit{ in } \mathbf{AK}_\mathbf{0}^= \right] \iff \tau =_\mu \sigma. \qquad (6.24)$$

Proof. The corollary is an immediate consequence of Theorem 6.3.18 and of Theorem 5.2.13, the correspondence theorem between $\mathbf{AK}_\mathbf{0}^=$-consistency and recursive type equality. □

Referring to the explanations we have given in Remark 5.2.7, (b), in Chapter 5, this corollary can now be viewed as a soundness and completeness theorem for the notion "existence of a consistency-unfolding in $\mathbf{AK}_\mathbf{0}^=$ " with respect to recursive type equality. The reason is that "existence of a consistency-unfolding in $\mathbf{AK}_\mathbf{0}^=$ " for an equation between recursive types is, as is easy to see, a positively calculable notion: there exists an effective positive test for it. Indeed, if for given $\tau, \sigma \in \mu Tp$ there exists a consistency-unfolding of $\tau = \sigma$ in $\mathbf{AK}_\mathbf{0}^=$, $\tau =_\mu \sigma$ follows by Theorem 6.3.13, and then a consistency-unfolding $\mathcal{C}$ of $\tau = \sigma$ in $\mathbf{AK}_\mathbf{0}^=$ can actually be found *effectively* by proceeding as suggested in the argumentation for the implication "$\Leftarrow$" in the proof sketch given above for Theorem 6.3.18.

6.4 Reflection Functions between Derivation-Trees in $\mathbf{AK}_{\mathbf{0}}^{=}$ and Derivations in e-$\mathbf{HB}_{\mathbf{0}}^{=}$

Having established a formal basis for the treatment of derivation-trees in $\mathbf{AK}_{\mathbf{0}}^{=}$ in the previous section, we are now able to formalize reflection operations of the kind considered in Section 6.1 between derivation-trees in $\mathbf{AK}_{\mathbf{0}}^{=}$ and derivations in $\mathbf{HB}_{\mathbf{0}}^{=}$. More precisely, we define in this section a pair of reflection functions between f.t.o.c.'s with b.l.o.m.f. in $\mathbf{AK}_{\mathbf{0}}^{=}$ and "pseudo-derivations" in $\mathbf{HB}_{\mathbf{0}}^{=}$.

The possibility to define such reflection functions rests on the fact that there exists a "duality" between rules of $\mathbf{HB}_{\mathbf{0}}^{=}$ and of $\mathbf{AK}_{\mathbf{0}}^{=}$. In particular, for every one-premise rule R of $\mathbf{HB}_{\mathbf{0}}^{=}$ there exists a one-premise rule R' of $\mathbf{AK}_{\mathbf{0}}^{=}$ (and vice versa) such that every instance of R corresponds to an instance of R' via exchanging the roles of premise and conclusion. For example, every instance of the rule FOLD$_l$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ corresponds to an instance of the rule UNFOLD$_l$ in $\mathbf{AK}_{\mathbf{0}}^{=}$, in the sense that, for all $\tau, \sigma \in \mu Tp$, $\alpha \in TVar$ and $\mathcal{D}_1 \in \mathcal{D}er(\mathbf{HB}_{\mathbf{0}}^{=})$, the application

$$\mathrm{FOLD}_l\ \dfrac{\begin{array}{c}\mathcal{D}_1\\ \tau[\mu\alpha.\,\tau/\alpha] = \sigma\end{array}}{\mu\alpha.\,\tau = \sigma} \qquad \text{corresponds to} \qquad \dfrac{\mu\alpha.\,\tau = \sigma}{\tau[\mu\alpha.\,\tau/\alpha] = \sigma}\ \mathrm{UNFOLD}_l$$

And furthermore, there is also an obvious correspondence between instances of the composition-rule ARROW in $\mathbf{HB}_{\mathbf{0}}^{=}$ and decomposition-branchings DECOMP in f.t.o.c.'s with b.l.o.m.f. in $\mathbf{AK}_{\mathbf{0}}^{=}$, that is, for all $\tau_1, \tau_2, \sigma_1, \sigma_2 \in \mu Tp$, the application

$$\mathrm{ARROW}\ \dfrac{\tau_1 = \sigma_1 \qquad \tau_2 = \sigma_2}{\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2} \qquad \text{corresponds to} \qquad \begin{array}{c}\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2\\ \hdashline \tau_2 = \sigma_2 \qquad \tau_1 = \sigma_1\end{array}\ \mathrm{DECOMP}$$

For the purpose of using this duality in transformations of prooftrees and of derivation-trees below, we introduce the following notation.

Notation 6.4.1 (Dual Rules in $\mathbf{HB}_{\mathbf{0}}^{=}$ and $\mathbf{AK}_{\mathbf{0}}^{=}$). We agree the following use of the syntactical variables $R^{(d)}$ and $R^{(cu)}$ for rules in, respectively, $\mathbf{HB}_{\mathbf{0}}^{=}$ and $\mathbf{AK}_{\mathbf{0}}^{=}$ to denote a bijective relationship between one-premise rules in $\mathbf{HB}_{\mathbf{0}}^{=}$ and rules in $\mathbf{AK}_{\mathbf{0}}^{=}$ different from DECOMP. This relationship is given by the below table

Rule $R^{(d)}$ in $\mathbf{HB}_{\mathbf{0}}^{=}$	FOLD$_l$	FOLD$_r$	REN	$(\mu{-}\bot)_l^{\bot\mathrm{der}}$	$(\mu{-}\bot)_r^{\bot\mathrm{der}}$
Rule $R^{(cu)}$ in $\mathbf{AK}_{\mathbf{0}}^{=}$	UNFOLD$_l$	UNFOLD$_r$	REN	$(\mu{-}\bot)_l^{\mathrm{der}\bot}$	$(\mu{-}\bot)_r^{\mathrm{der}\bot}$

and our intended use of $R^{(d)}$ and $R^{(cu)}$ is one that is explained by the following example: if in a particular context (of a proof, an argument, etc.) the syntactical variable $R^{(d)}$ is used for the rule FOLD$_r$ in $\mathbf{HB}_{\mathbf{0}}^{=}$, then $R^{(cu)}$ will stand in the same context for the rule UNFOLD$_r$ in $\mathbf{AK}_{\mathbf{0}}^{=}$.

For all pairs $\langle R^{(d)}, R^{(cu)}\rangle$ of rules that appear in the same column of the above table, we say that $R^{(d)}$ and $R^{(cu)}$ are *dual* rules, that $R^{(d)}$ is the *dual rule of* $R^{(cu)}$ *in* $\mathbf{HB}_{\mathbf{0}}^{=}$ and that $R^{(cu)}$ is the *dual rule of* $R^{(d)}$ *in* $\mathbf{AK}_{\mathbf{0}}^{=}$. ⊠

With this notation, we are able formulate that part of the duality between rules of $\mathbf{AK}_{\mathbf{0}}^{=}$ and $\mathbf{HB}_{\mathbf{0}}^{=}$ which concerns one-premise rules of $\mathbf{HB}_{\mathbf{0}}^{=}$ in the following remark.

Remark 6.4.2 (Duality between rules of $\mathbf{HB}_{\mathbf{0}}^{=}$ and $\mathbf{AK}_{\mathbf{0}}^{=}$). The following two statements are true for the correspondence according to Notation 6.4.1 between dual rules of $\mathbf{AK}_{\mathbf{0}}^{=}$ and $\mathbf{HB}_{\mathbf{0}}^{=}$:

(i) For all one-premise rules $R^{(d)}$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ it is the case that:

$$\left.\begin{array}{l} \text{For all } \tau,\sigma,\tau_1,\sigma_1 \in \mu Tp, \text{ and all derivations } \mathcal{D}_1 \text{ in } \mathbf{HB}_{\mathbf{0}}^{=} : \\[1ex] \dfrac{\begin{array}{c}\mathcal{D}_1\\ \tau_1=\sigma_1\end{array}}{\tau=\sigma}\,R^{(d)} \quad \text{is a rule application in } \mathbf{HB}_{\mathbf{0}}^{=} \quad \Longleftrightarrow \\[2ex] \qquad \Longleftrightarrow \quad \dfrac{\tau=\sigma}{\tau_1=\sigma_1}\,R^{(cu)} \quad \text{is a rule application in } \mathbf{AK}_{\mathbf{0}}^{=}. \end{array}\right\} \qquad (6.25)$$

(ii) Also, (6.25) holds for all rules $R^{(cu)}$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ *except* the rule DECOMP.

The reflection functions, which are to be defined below, are going to map f.t.o.c.'s with b.l.o.m.f. to a generalization of derivations in **e-**$\mathbf{HB}_{\mathbf{0}}^{=}$, to be called "pseudo-derivations in **e-**$\mathbf{HB}_{\mathbf{0}}^{=}$", and vice versa. We therefore need to define "pseudo-derivations in **e-**$\mathbf{HB}_{\mathbf{0}}^{=}$" first.

By a *pseudo-derivation in* **e-**$\mathbf{HB}_{\mathbf{0}}^{=}$ we understand a prooftree that differs from a derivation in **e-**$\mathbf{HB}_{\mathbf{0}}^{=}$ insofar as that the side-conditions $\mathbf{C}$ do not have to be satisfied for all applications of R/FIX-rules in $\boldsymbol{\mathcal{D}}(\mathcal{C})$, where $R \in \{\text{FOLD}_{l/r}, \text{REN}\}$. Or equivalently, a pseudo-derivation in **e-**$\mathbf{HB}_{\mathbf{0}}^{=}$ is a derivation in a variant system $(\textbf{e-}\mathbf{HB}_{\mathbf{0}}^{=})'$ of **e-**$\mathbf{HB}_{\mathbf{0}}^{=}$ that has the same formulas, axioms, and rules as **e-**$\mathbf{HB}_{\mathbf{0}}^{=}$, but where the side-condition $\mathbf{C}$ on the contractiveness for immediate subderivations is dropped as requirement for applications of rules REN/FIX and $\text{FOLD}_{l/r}$/FIX.

We want to mention that pseudo-derivations in **e-**$\mathbf{HB}_{\mathbf{0}}^{=}$ as just defined are special cases of "pseudo-derivations" in **e-**$\mathbf{HB}_{\mathbf{0}}^{=}$ according to Definition B.2.13 on page 381 in Appendix B (which, roughly speaking, are prooftrees with inferences labelled by **e-**$\mathbf{HB}_{\mathbf{0}}^{=}$-rules such that the inferences do not even have to be correct applications of the rules by which they are labelled). With respect to this later defined notion of "pseudo-derivation", pseudo-derivations in $\mathbf{HB}_{\mathbf{0}}^{=}$ as defined here could be called "pseudo-derivations in **e-**$\mathbf{HB}_{\mathbf{0}}^{=}$ with respect to the side-condition $\mathbf{C}$ on applications of $\cdot$/FIX-rules". However, since the concept of "pseudo-derivation in **e-**$\mathbf{HB}_{\mathbf{0}}^{=}$" as defined above is vital, but confined to this chapter, and because we do not need here the concept "pseudo-derivation in **e-**$\mathbf{HB}_{\mathbf{0}}^{=}$" as derived from Definition B.2.13, we have chosen to use this term in the more restricted sense in this chapter.

Definition and Lemma 6.4.3. (Reflection functions between f.t.o.c.'s with b.l.o.m.f. in $\mathbf{AK}_{\mathbf{0}}^{=}$ and pseudo-derivations in e-$\mathbf{HB}_{\mathbf{0}}^{=}$). In items (i) and (ii) below we respectively define the *reflection function* $\boldsymbol{\mathcal{D}}$ that maps f.t.o.c.'s with b.l.o.m.f. in $\mathbf{AK}_{\mathbf{0}}^{=}$ to pseudo-derivations in **e-**$\mathbf{HB}_{\mathbf{0}}^{=}$, and the *reflection function* $\boldsymbol{\mathcal{C}}$ that maps pseudo-derivations in **e-**$\mathbf{HB}_{\mathbf{0}}^{=}$ to f.t.o.c.'s with b.l.o.m.f. in $\mathbf{AK}_{\mathbf{0}}^{=}$.

(i) For every f.t.o.c. with b.l.o.m.f. $\mathcal{C}$ from $\tau = \sigma$ in $\mathbf{AK_0^{=}}$, where $\tau, \sigma \in \mu Tp$, the *reflection* $\boldsymbol{\mathcal{D}}(\mathcal{C})$ of $\mathcal{C}$ is defined by induction on the depth $|\mathcal{C}|$ of $\mathcal{C}$ according to the five clauses detailed in Figure 6.5 that arise by case-distinction dependent on the last step of the generation of $\mathcal{C}$ according to Definition 6.3.10.

The difference in the cases of f.t.o.c.'s with b.l.o.m.f., to which the second and the third, and respectively, the fourth and the fifth inductive clause in Figure 6.5 apply, consists in whether or not the formula at the root of the considered derivation-tree is marked or not. This formula is assumed to carry a marker in the third and in the fifth clause, whereas it is unmarked in the second and in the fourth clause.

(ii) For every pseudo-derivation $\mathcal{D}$ in $\mathbf{e\text{-}HB_0^{=}}$ with conclusion $\tau = \sigma$, where $\tau, \sigma \in$ $\in \mu Tp$, the *reflection* $\boldsymbol{\mathcal{C}}(\mathcal{D})$ of $\mathcal{D}$ is a f.t.o.c. with b.l.o.m.f. from $\tau = \sigma$ in $\mathbf{AK_0^{=}}$ that is defined by induction on the depth $|\mathcal{D}|$ of $\mathcal{D}$ according to the five clauses gathered in Figure 6.6.

⊠

Proof. We want to show the well-definedness of the reflection functions $\boldsymbol{\mathcal{D}}$ and $\boldsymbol{\mathcal{C}}$ defined above, thereby proving the 'lemma-part' of Definition and Lemma 6.4.3.

(a) For the well-definedness of the reflection function $\boldsymbol{\mathcal{D}}$ according to the formulation in Definition 6.4.3 (i) there are two assertions that have to be confirmed:

(I) The reflection function $\boldsymbol{\mathcal{D}}$ is in fact defined for all f.t.o.c.'s with b.l.o.m.f. $\mathcal{C}$ in $\mathbf{AK_0^{=}}$.

(II) For every f.t.o.c. with b.l.o.m.f. $\mathcal{C}$ of $\tau = \sigma$ in $\mathbf{AK_0^{=}}$ (with some $\tau, \sigma \in \mu Tp$) its reflection $\boldsymbol{\mathcal{D}}(\mathcal{C})$ is a "pseudo-derivation" of $\tau = \sigma$ in $\mathbf{e\text{-}HB_0^{=}}$, i.e. a derivation-tree of the shape of a derivation in $\mathbf{e\text{-}HB_0^{=}}$, in which (only) the side-conditions $\mathbf{C}$ on applications of FOLD$_{l/r}$/FIX and REN/FIX may be violated.

The validity of the statement in (I), i.e. that $\mathcal{D}$ is in fact defined for all f.t.o.c.'s with b.l.o.m.f. in $\mathbf{AK_0^{=}}$, can be recognized from the fact that the 5 inductive clauses in the definition of $\boldsymbol{\mathcal{D}}$ in Figure 6.5 cover exactly all 6 clauses in Definition 6.3.10, the alternative inductive definition for f.t.o.c.'s with b.l.o.m.f. in $\mathbf{AK_0^{=}}$ (hereby the first inductive clause in Figure 6.5 covers f.t.o.c.'s with b.l.o.m.f.'s of the base cases (i) and (ii) in Definition 6.3.10).

To demonstrate the fulfilledness of assertion (II) it suffices to show for every f.t.o.c. with b.l.o.m.f. $\mathcal{C}$ of $\tau = \sigma$ in $\mathbf{AK_0^{=}}$ (with some $\tau, \sigma \in \mu Tp$) that the side-conditions $\mathbf{I}$ are satisfied for all applications of FOLD$_{l/r}$/FIX and REN/FIX as well as for all applications of ARROW/FIX occurring in $\boldsymbol{\mathcal{D}}(\mathcal{C})$.

This can be shown by an easy induction on the depth $|\mathcal{C}|$ of an arbitrary given f.t.o.c. $\mathcal{C}$ with b.l.o.m.f. in $\mathbf{AK_0^{=}}$, in which induction also the statement is shown, that $\boldsymbol{\mathcal{D}}(\mathcal{C})$ always has the same inhabited classes of open marked assumptions as $\mathcal{C}$ has classes of unbound leaf-occurrences of marked formulas.

Figure 6.5: Inductive definition of the reflection function $\boldsymbol{\mathcal{D}}$ that maps f.t.o.c. with b.l.o.m.f.'s $\mathcal{C}$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ to pseudo-derivations $\boldsymbol{\mathcal{D}}(\mathcal{C})$ in $\mathbf{e}\text{-}\mathbf{HB}_{\mathbf{0}}^{=}$:

$$\boxed{(\tau = \sigma)^{\boldsymbol{m}}} \quad \overset{\boldsymbol{\mathcal{D}}}{\longmapsto} \quad \boxed{(\tau = \sigma)^{\boldsymbol{m}}}$$

$$\boxed{\begin{array}{c} \dfrac{\tau = \sigma}{(\tau_1 = \sigma_1)^{\boldsymbol{m}_1}}\, R^{(cu)} \\ \mathcal{C}_1 \end{array}} \quad \overset{\boldsymbol{\mathcal{D}}}{\longmapsto} \quad \boxed{\begin{array}{c} \boldsymbol{\mathcal{D}}(\mathcal{C}_1) \\ \dfrac{\tau_1 = \sigma_1}{\tau = \sigma}\, R^{(d)} \end{array}}$$

(for $R^{(cu)} \in \{\text{UNFOLD}_{l/r},\ \text{REN},\ (\mu{-}\bot)^{\text{der}\bot}_{l/r}\}$)

$$\boxed{\begin{array}{c} \dfrac{(\tau = \sigma)^{u}}{(\tau_1 = \sigma_1)^{\boldsymbol{m}_1}}\, R^{(cu)} \\ \mathcal{C}_1 \\ [\tau = \sigma]^{u} \end{array}} \quad \overset{\boldsymbol{\mathcal{D}}}{\longmapsto} \quad \boxed{\begin{array}{c} [\tau = \sigma]^{u} \\ \boldsymbol{\mathcal{D}}(\mathcal{C}_1) \\ \dfrac{\tau_1 = \sigma_1}{\tau = \sigma}\, R^{(d)}/\text{FIX},\, u \end{array}}$$

(for $R^{(cu)} \in \{\text{UNFOLD}_{l/r},\ \text{REN}\}$)

$$\boxed{\begin{array}{cc} \multicolumn{2}{c}{\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2} \\ \hdashline (\tau_1 = \sigma_1)^{\boldsymbol{m}_1} & (\tau_2 = \sigma_2)^{\boldsymbol{m}_2} \\ \mathcal{C}_1 & \mathcal{C}_2 \end{array}\ \text{DECOMP}} \quad \overset{\boldsymbol{\mathcal{D}}}{\longmapsto}$$

$$\overset{\boldsymbol{\mathcal{D}}}{\longmapsto} \quad \boxed{\begin{array}{cc} \boldsymbol{\mathcal{D}}(\mathcal{C}_1) & \boldsymbol{\mathcal{D}}(\mathcal{C}_2) \\ \tau_1 = \sigma_1 & \tau_2 = \sigma_2 \\ \hline \multicolumn{2}{c}{\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2} \end{array}\ \text{ARROW}}$$

$$\boxed{\begin{array}{cc} \multicolumn{2}{c}{(\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2)^{u}} \\ \hdashline (\tau_1 = \sigma_1)^{\boldsymbol{m}_1} & (\tau_2 = \sigma_2)^{\boldsymbol{m}_2} \\ \mathcal{C}_1 & \mathcal{C}_2 \\ [\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2]^{u} & [\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2]^{u} \end{array}\ \text{DECOMP}} \quad \overset{\boldsymbol{\mathcal{D}}}{\longmapsto}$$

$$\overset{\boldsymbol{\mathcal{D}}}{\longmapsto} \quad \boxed{\begin{array}{cc} [\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2]^{u} & [\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2]^{u} \\ \boldsymbol{\mathcal{D}}(\mathcal{C}_1) & \boldsymbol{\mathcal{D}}(\mathcal{C}_2) \\ \tau_1 = \sigma_1 & \tau_2 = \sigma_2 \\ \hline \multicolumn{2}{c}{\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2} \end{array}\ \text{ARROW/FIX},\, u}$$

Figure 6.6: Inductive definition of the reflection function $\boldsymbol{\mathcal{C}}$ that maps pseudo-derivations $\mathcal{D}$ in $\mathbf{e\text{-}HB}_{\mathbf{0}}^{=}$ to f.t.o.c. with b.l.o.m.f.'s $\boldsymbol{\mathcal{C}}(\mathcal{D})$ in $\mathbf{AK}_{\mathbf{0}}^{=}$:

$$\boxed{(\tau=\sigma)^{\boldsymbol{m}}} \;\overset{\boldsymbol{\mathcal{C}}}{\longmapsto}\; \boxed{(\tau=\sigma)^{\boldsymbol{m}}}$$

$$\boxed{\dfrac{\begin{array}{c}\mathcal{D}_1\\ \tau_1=\sigma_1\end{array}}{\tau=\sigma}\,R^{(d)}} \;\overset{\boldsymbol{\mathcal{C}}}{\longmapsto}\; \boxed{\begin{array}{c}\dfrac{\tau=\sigma}{(\tau_1=\sigma_1)^{\boldsymbol{m}_1}}\,R^{(cu)}\\ \boldsymbol{\mathcal{C}}(\mathcal{D}_1)\end{array}}$$

(for $R^{(d)} \in \{\text{FOLD}_{l/r}, \text{REN}, (\mu\text{−}\bot)^{\bot\text{der}}_{l/r}\}$)

$$\boxed{\dfrac{\begin{array}{c}[\tau=\sigma]^u\\ \mathcal{D}_1\\ \tau_1=\sigma_1\end{array}}{\tau=\sigma}\,R^{(d)}/\text{FIX},\,u} \;\overset{\boldsymbol{\mathcal{C}}}{\longmapsto}\; \boxed{\begin{array}{c}\dfrac{(\tau=\sigma)^u}{(\tau_1=\sigma_1)^{\boldsymbol{m}_1}}\,R^{(cu)}\\ \boldsymbol{\mathcal{C}}(\mathcal{D}_1)\\ [\tau=\sigma]^u\end{array}}$$

(for $R^{(d)} \in \{\text{FOLD}_{l/r}, \text{REN}\}$)

$$\boxed{\dfrac{\begin{array}{c}\mathcal{D}_1\\ \tau_1=\sigma_1\end{array} \qquad \begin{array}{c}\mathcal{D}_2\\ \tau_2=\sigma_2\end{array}}{\tau_1\to\tau_2=\sigma_1\to\sigma_2}\,\text{ARROW}} \;\overset{\boldsymbol{\mathcal{C}}}{\longmapsto}$$

$$\overset{\boldsymbol{\mathcal{C}}}{\longmapsto}\; \boxed{\begin{array}{cc}\multicolumn{2}{c}{\tau_1\to\tau_2=\sigma_1\to\sigma_2}\\ \hdashline (\tau_1=\sigma_1)^{\boldsymbol{m}_1} & (\tau_2=\sigma_2)^{\boldsymbol{m}_2}\\ \boldsymbol{\mathcal{C}}(\mathcal{D}_1) & \boldsymbol{\mathcal{C}}(\mathcal{D}_2)\end{array}\;\text{DECOMP}}$$

$$\boxed{\dfrac{\begin{array}{c}[\tau_1\to\tau_2=\sigma_1\to\sigma_2]^u\\ \mathcal{D}_1\\ \tau_1=\sigma_1\end{array} \qquad \begin{array}{c}[\tau_1\to\tau_2=\sigma_1\to\sigma_2]^u\\ \mathcal{D}_2\\ \tau_2=\sigma_2\end{array}}{\tau_1\to\tau_2=\sigma_1\to\sigma_2}\,\text{ARROW/FIX},\,u} \;\overset{\boldsymbol{\mathcal{C}}}{\longmapsto}$$

$$\overset{\boldsymbol{\mathcal{C}}}{\longmapsto}\; \boxed{\begin{array}{cc}\multicolumn{2}{c}{(\tau_1\to\tau_2=\sigma_1\to\sigma_2)^u}\\ \hdashline (\tau_1=\sigma_1)^{\boldsymbol{m}_1} & (\tau_2=\sigma_2)^{\boldsymbol{m}_2}\\ \boldsymbol{\mathcal{C}}(\mathcal{D}_1) & \boldsymbol{\mathcal{C}}(\mathcal{D}_2)\\ [\tau_1\to\tau_2=\sigma_1\to\sigma_2]^u & [\tau_1\to\tau_2=\sigma_1\to\sigma_2]^u\end{array}\;\text{DECOMP}}$$

This auxiliary statement can be visualized as the part concerning the application of the function $\boldsymbol{\mathcal{D}}$ in the following picture, i.e. the transition from left to right in:

$$\boxed{\begin{array}{c}(\tau=\sigma)^{\boldsymbol{m}}\\ \mathcal{C}\\ \{\,[\tau_i=\sigma_i]^{u_i}\,\}_{i=1,\ldots,n}\end{array}} \quad \begin{array}{c}\xmapsto{\boldsymbol{\mathcal{D}}}\\ \xleftarrow[\boldsymbol{\mathcal{C}}]{}\end{array} \quad \boxed{\begin{array}{c}\{\,[\tau_i=\sigma_i]^{u_i}\,\}_{i=1,\ldots,n}\\ \mathcal{D}\\ \tau=\sigma\end{array}} \tag{6.26}$$

Hereby the displayed family $\{\,[\tau_i=\sigma_i]^{u_i}\,\}_{i=1,\ldots,n}$ (with $n \in \omega$, $\tau_i, \sigma_i \in \mu Tp$ and u_i a marker for $i = 1,\ldots,n$) gathers in $\mathcal{C}$ precisely all inhabited classes of unbound leaf-occurrences of marked formulas, and assembles in $\mathcal{D}$ precisely all inhabited open assumption classes.

(b) Since the inductive clauses for the definition of the reflection function $\boldsymbol{\mathcal{C}}$ in Figure 6.6 cover all cases of axioms, assumptions and of last rule applications in $\mathbf{e\text{-}HB}_{\mathbf{0}}^{=}$-derivations $\mathcal{D}$, $\boldsymbol{\mathcal{C}}$ is in fact defined for all $\mathbf{e\text{-}HB}_{\mathbf{0}}^{=}$-derivations. Hence for the well-definedness of the reflection function $\boldsymbol{\mathcal{C}}$ it has to be shown, for all $\mathbf{e\text{-}HB}_{\mathbf{0}}^{=}$-derivations $\mathcal{D}$ of $\tau = \sigma$ (where $\tau, \sigma \in \mu Tp$), that its reflection $\mathcal{C} = \boldsymbol{\mathcal{C}}(\mathcal{D})$ is in fact a f.t.o.c. with b.l.o.m.f. from $\tau = \sigma$ in $\mathbf{AK}_{\mathbf{0}}^{=}$. More precisely, it has to be established that each occurrence of a marked formula $(\rho = \chi)^u$ (for some $\rho, \chi \in \mu Tp$ and a marker u) *inside* $\mathcal{C}$ binds at least one occurrence of $(\rho = \chi)^u$ at a leaf-position in $\mathcal{C}$.

This can again be shown in a straightforward manner by induction on the depth $|\mathcal{D}|$ of $\mathcal{D}$, when it is shown at the same time (i.e. as part of the statement to prove by this induction) that $\mathcal{C}$ always possesses the same inhabited classes of unbound leaf-occurrences of marked formulas as $\mathcal{D}$ has as inhabited classes of open marked assumptions. This is illustrated by the part concerning the function $\boldsymbol{\mathcal{C}}$ in picture (6.26), i.e. the transition from right to left there.

□

6.5 A Duality between Consistency-Unfoldings in $\mathbf{AK}_{\mathbf{0}}^{=}$ and Derivations in $\mathbf{e\text{-}HB}_{\mathbf{0}}^{=}$

Due to the preparatory work of the previous sections, we are finally in possession of all formal concepts necessary to formulate our initial observation about a relationship between the systems $\mathbf{HB}_{\mathbf{0}}^{=}$ and $\mathbf{AK}_{\mathbf{0}}^{=}$ into a precise statement. In the present section we therefore state and prove the main theorem of this chapter, and give an illustration of this theorem by an easy example.

This theorem states the existence of a very immediate correspondence between derivations in $\mathbf{e\text{-}HB}_{\mathbf{0}}^{=}$ without open assumptions and consistency-unfoldings in $\mathbf{AK}_{\mathbf{0}}^{=}$. Hereby the correspondence takes place via the reflection functions defined in the previous section and can geometrically be visualized in all of its instances (for this we will see an example below). This is reason for us to speak of a *duality* between derivations in $\mathbf{e\text{-}HB}_{\mathbf{0}}^{=}$ without open assumptions and consistency-unfoldings in $\mathbf{AK}_{\mathbf{0}}^{=}$.

Theorem 6.5.1 (A duality between derivations in e-HB$_{\mathbf{0}}^{=}$ and consistency-unfoldings in AK$_{\mathbf{0}}^{=}$). *There is a bijective functional relationship between derivations in* $\mathbf{e\text{-}HB}_{\mathbf{0}}^{=}$ *without open assumption classes and consistency-unfoldings in* $\mathbf{AK}_{\mathbf{0}}^{=}$ *that takes place via the reflection functions* $\boldsymbol{\mathcal{D}}$ *and* $\boldsymbol{\mathcal{C}}$ *defined in Definition 6.4.3. More precisely, the following three assertions hold:*

(i) For every derivation $\mathcal{D}$ *in* $\mathbf{e\text{-}HB}_{\mathbf{0}}^{=}$ *without open assumption classes and with conclusion* $\tau = \sigma$ *(for some* $\tau, \sigma \in \mu Tp$*) its reflection* $\boldsymbol{\mathcal{C}}(\mathcal{D})$ *is a consistency-unfolding of* $\tau = \sigma$ *in* $\mathbf{AK}_{\mathbf{0}}^{=}$*.*

(ii) For every consistency-unfolding $\mathcal{C}$ *of* $\tau = \sigma$ *in* $\mathbf{AK}_{\mathbf{0}}^{=}$ *(with some* $\tau, \sigma \in \mu Tp$*) its reflection* $\boldsymbol{\mathcal{D}}(\mathcal{C})$ *is a derivation in* $\mathbf{e\text{-}HB}_{\mathbf{0}}^{=}$ *with conclusion* $\tau = \sigma$ *and without open assumption classes.*

(iii) The restrictions

$$\begin{aligned} \boldsymbol{\mathcal{D}}|_{\mathcal{D}er_{\emptyset}(\mathbf{e\text{-}HB}_{\mathbf{0}}^{=})} &: \mathcal{D}er_{\emptyset}(\mathbf{e\text{-}HB}_{\mathbf{0}}^{=}) \longrightarrow CU(\mathbf{AK}_{\mathbf{0}}^{=}), \quad \textit{and} \\ \boldsymbol{\mathcal{C}}|_{CU(\mathbf{AK}_{\mathbf{0}}^{=})} &: CU(\mathbf{AK}_{\mathbf{0}}^{=}) \longrightarrow \mathcal{D}er_{\emptyset}(\mathbf{e\text{-}HB}_{\mathbf{0}}^{=}) \end{aligned}$$

of the reflection functions $\boldsymbol{\mathcal{D}}$ *and* $\boldsymbol{\mathcal{C}}$ *to, respectively, the set* $\mathcal{D}er_{\emptyset}(\mathbf{e\text{-}HB}_{\mathbf{0}}^{=})$ *of derivations in* $\mathbf{AK}_{\mathbf{0}}^{=}$ *without open assumptions, and to the set* $CU^{\mathbf{D}}(\mathbf{AK}_{\mathbf{0}}^{=})$ *of consistency-unfoldings in* $\mathbf{AK}_{\mathbf{0}}^{=}$ *are inverses of each other.*

Proof. (a) The statement in item (i) of the lemma is a consequence of the following more general statement, which asserts that the reflection function $\boldsymbol{\mathcal{C}}$ maps $\mathbf{e\text{-}HB}_{\mathbf{0}}^{=}$-derivations with (possibly) open assumption classes to partial consistency-unfoldings in $\mathbf{AK}_{\mathbf{0}}^{=}$:

For every $\mathbf{e\text{-}HB}_{\mathbf{0}}^{=}$*-derivation* $\mathcal{D}$ with conclusion $\tau = \sigma$, whose open marked assumptions are precisely those that belong to one of the assumption classes $[\tau_i = \sigma_i]^{u_i}$ for $i \in \{1, \ldots, n\}$ (with some $n \in \omega$), the reflection function $\boldsymbol{\mathcal{C}}$ maps $\mathcal{D}$ to a *partial consistency-unfolding* $\boldsymbol{\mathcal{C}}(\mathcal{D})$ of $\tau = \sigma$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ in a way which can be illustrated as:

$$\boxed{\begin{array}{c} \{\, [\tau_i = \sigma_i]^{u_i} \,\}_{i=1,\ldots,n} \\ \mathcal{D} \\ \tau = \sigma \end{array}} \quad \overset{\boldsymbol{\mathcal{C}}}{\longmapsto} \quad \boxed{\begin{array}{c} (\tau = \sigma)^{\boldsymbol{m}} \\ \boldsymbol{\mathcal{C}}(\mathcal{D}) \\ \{\, [\tau_i = \sigma_i]^{u_i} \,\}_{i=1,\ldots,n} \end{array}}$$

(where according to the notation introduced in Definition 6.4.3 the occurrence of $(\tau = \sigma)^{\boldsymbol{m}}$ at the root of $\boldsymbol{\mathcal{C}}(\mathcal{D})$ stands either for the unmarked equation $\tau = \sigma$ or for a marked formula $(\tau = \sigma)^{v}$ with some assumption marker v that is assumed to be the designation of the syntactical variable $\boldsymbol{m}$ in this case). Hereby the notation used for $\boldsymbol{\mathcal{C}}(\mathcal{D})$ is to be understood as follows. The partial consistency-unfolding (and hence f.t.o.c. with b.l.o.m.f.) $\boldsymbol{\mathcal{C}}(\mathcal{D})$ possesses in its leaves at the bottom occurrences of marked formulas $(\tau_i = \sigma_i)^{u_i}$ for all $i \in \{1, \ldots, n\}$ that are *not bound back* in $\boldsymbol{\mathcal{C}}(\mathcal{D})$, which implies in particular that no such marked formula $(\tau_i = \sigma_i)^{u_i}$ is bound back in $\boldsymbol{\mathcal{C}}(\mathcal{D})$ to the root-occurrence of $(\tau = \sigma)^{\boldsymbol{m}}$ of $\boldsymbol{\mathcal{C}}(\mathcal{D})$. Furthermore it holds that each unbound leaf-occurrence of a marked formula $(\tilde{\tau} = \tilde{\sigma})^{\tilde{u}}$ in $\boldsymbol{\mathcal{C}}(\mathcal{D})$ must be a member of one of the displayed classes $[\tau_i = \sigma_i]^{u_i}$ for $i \in \{1, \ldots, n\}$ of formulas not bound back in $\boldsymbol{\mathcal{C}}(\mathcal{D})$, i.e. it must hold that $\tilde{\tau} \equiv \tau_i$, $\tilde{\sigma} \equiv \sigma_i$ and $\tilde{u} \equiv u$ for some $i \in \{1, \ldots, n\}$. And moreover it holds, that whenever $\mathcal{D}$ contains at least one application of a rule ARROW or ARROW/FIX, then $\boldsymbol{\mathcal{C}}(\mathcal{D})$ contains at least one occurrence of a branching DECOMP.

This statement, which is more general than item (i) of the lemma, can be shown, for every derivation $\mathcal{D}$ in $\mathbf{e\text{-}HB}_{\mathbf{0}}^{=}$, by induction on $|\mathcal{D}|$.

For the base case of the induction we notice that both every axiom (REFL) $\tau = \tau$ and every open marked assumption $(\tau = \sigma)^u$ of $\mathbf{e\text{-}HB}_{\mathbf{0}}^{=}$ are mapped by $\boldsymbol{\mathcal{C}}$ to themselves respectively. From this it is obvious to see that the above statement holds in both of these cases.

For the induction step we will only consider the case in which the last rule application in $\mathcal{D}$ is $R^{(d)}$/FIX for some rule $R^{(d)} \in \{\text{FOLD}_l, \text{FOLD}_r, \text{REN}\}$, i.e. that $\mathcal{D}$ is an $\mathbf{e\text{-}HB}_{\mathbf{0}}^{=}$-derivation of the form

$$\begin{array}{c} [\tau = \sigma]^u \;\; \{[\tau_i = \sigma_i]^{u_i}\}_{i=1,\ldots,n} \\ \mathcal{D}_1 \\ \dfrac{\tilde{\tau}_1 = \tilde{\sigma}_1}{\tau = \sigma}\, R^{(d)}/\text{FIX},\, u \end{array}$$

for an $n \in \omega$ and some $\tau, \sigma, \tilde{\tau}_1, \tilde{\sigma}_1, \tau_1, \sigma_1, \ldots, \tau_n, \sigma_n \in \mu Tp$ as well as with open assumption classes $[\tau_i = \sigma_i]^{u_i}$ for $i \in \{1, \ldots, n\}$. We notice that, due to the side-condition $\mathbf{I}$ on the application of $R^{(d)}$/FIX at the bottom of $\mathcal{D}$, the open assumption class $[\tau = \sigma]^u$ in $\mathcal{D}_1$, which consists of the assumptions that are discharged at the bottommost rule application in $\mathcal{D}$, must be inhabited. Performing the induction step for this case is not just as straightforward as are all cases of $\mathbf{e\text{-}HB}_{\mathbf{0}}^{=}$-derivations with occurrences of other rules as last rule application in $\mathcal{D}$; these cases can be treated in a similar and rather easier way. In the situation here, we find as a consequence of the induction hypothesis that $\boldsymbol{\mathcal{C}}(\mathcal{D}_1)$ is a partial consistency-unfolding of $\tilde{\tau}_1 = \tilde{\sigma}_1$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ that contains precisely the classes $[\tau = \sigma]^u$ and $[\tau_i = \sigma_i]^{u_i}$ for $i \in \{1, \ldots, n\}$ of such marked formulas in leaves at the bottom of $\boldsymbol{\mathcal{C}}(\mathcal{D}_1)$ that are not bound back in $\boldsymbol{\mathcal{C}}(\mathcal{D}_1)$. It follows that the derivation-tree $\boldsymbol{\mathcal{C}}(\mathcal{D})$

$$\begin{array}{c} \dfrac{(\tau = \sigma)^u}{(\tilde{\tau}_1 = \tilde{\sigma}_1)^{\boldsymbol{m}_1}}\, R^{(cu)} \\ \boldsymbol{\mathcal{C}}(\mathcal{D}_1) \\ [\tau = \sigma]^u \;\; \{[\tau_i = \sigma_i]^{u_i}\}_{i=1,\ldots,n} \end{array}$$

the result of applying the reflection function $\boldsymbol{\mathcal{C}}$, to the derivation $\mathcal{D}$ according to the third clause in its inductive definition in Figure 6.6, to the derivation $\mathcal{D}$, is a f.t.o.c. with b.l.o.m.f. in $\mathbf{AK}_{\mathbf{0}}^{=}$ that contains as classes unbound leaf-occurrences of marked formulas exactly all classes $[\tau_i = \sigma_i]^{u_i}$ for $i \in \{1, \ldots, n\}$. It remains to be shown that $\boldsymbol{\mathcal{C}}(\mathcal{D})$ is in fact a partial consistency-unfolding of $\tau = \sigma$ in $\mathbf{AK}_{\mathbf{0}}^{=}$, i.e. we have to make sure that there is actually always an occurrence of a branching DECOMP in every thread in $\boldsymbol{\mathcal{C}}(\mathcal{D})$ that connects occurrences of marked formulas linked through backbindings.

For all leaf-occurrences of marked formulas in $\boldsymbol{\mathcal{C}}(\mathcal{D})$ that are bound back in $\boldsymbol{\mathcal{C}}(\mathcal{D})$, but that are not part of the displayed class $[\tau = \sigma]^u$ this condition follows already from the fact that $\boldsymbol{\mathcal{C}}(\mathcal{D}_1)$ is, due to the induction hypothesis, a partial consistency-unfolding. Hence it suffices to demonstrate the existence of an occurrence of a branching DECOMP in each thread in $\boldsymbol{\mathcal{C}}(\mathcal{D})$ from the root-occurrence of $(\tau = \sigma)^u$ down to a leaf-occurrence of $(\tau = \sigma)^u$ belonging to the displayed class $[\tau = \sigma]^u$ in $\boldsymbol{\mathcal{C}}(\mathcal{D})$. Or equivalently, it suffices to show the existence of just a single occurrence of a branching DECOMP in $\boldsymbol{\mathcal{C}}(\mathcal{D}_1)$: since due to the side-conditions **I** and **C** on the application of $R^{(d)}$/FIX at the bottom of $\mathcal{D}$ there must at least be one application of a rule ARROW or ARROW/FIX in $\mathcal{D}_1$, it follows by the induction hypothesis (referring to the sentence beginning with "And moreover ..." at the end of the statement to show) that there must then also be at least one occurrence of a branching DECOMP in $\boldsymbol{\mathcal{C}}(\mathcal{D}_1)$ and hence also in $\boldsymbol{\mathcal{C}}(\mathcal{D})$. Hence a branching DECOMP is always crossed in each thread in $\boldsymbol{\mathcal{C}}(\mathcal{D})$ that leads from the root of $\boldsymbol{\mathcal{C}}(\mathcal{D})$ down to a leaf-occurrence of $(\rho = \chi)^u$ in $\boldsymbol{\mathcal{C}}(\mathcal{D})$ that is bound back to the root-occurrence of $(\rho = \chi)^u$. Since we know that $\mathcal{D}_1$ contains at least one application of ARROW or ARROW/FIX and that $\mathcal{C}_1$ contains at least one occurrence of a branching DECOMP, these facts are also true with respect to $\mathcal{C}$ and respectively with respect to $\mathcal{D}$. Hence also the part with respect to the sentence "And moreover ..." of the statement to show for the induction step is shown.

Thus $\boldsymbol{\mathcal{C}}(\mathcal{D})$ is a partial consistency-unfolding of the particular form as required by the statement to prove for the induction step. Hereby we have succeeded in performing the induction step in the considered case.

(b) Quite analogously to the proof of item (i) above, the statement in item (ii) of the lemma is a consequence of the following more general statement.

The effect of performing the reflection function $\boldsymbol{\mathcal{D}}$ defined in Definition 6.4.3, (ii), on an arbitrary given partial consistency-unfolding $\mathcal{C}$ is described by the following picture:

$$\boxed{\begin{array}{c}(\tau = \sigma)^{\boldsymbol{m}}\\ \mathcal{C}\\ \{\,[\tau_i = \sigma_i]^{u_i}\,\}_{i=1,\ldots,n}\end{array}} \quad \overset{\boldsymbol{\mathcal{D}}}{\longmapsto} \quad \boxed{\begin{array}{c}\{\,[\tau_i = \sigma_i]^{u_i}\,\}_{i=1,\ldots,n}\\ \boldsymbol{\mathcal{D}}(\mathcal{C})\\ \tau = \sigma\end{array}}$$

This means: every *partial consistency-unfolding* $\mathcal{C}$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ of an equation $\tau = \sigma$ (for some $\tau, \sigma \in \mu Tp$) that possesses precisely the classes $[\tau = \sigma]^{u_i}$ for $i = 1, \ldots, n$ as classes of unbound leaf-occurrences of marked assumptions is mapped by the reflection function $\boldsymbol{\mathcal{D}}$ to an **e-HB**$_{\mathbf{0}}^{=}$-derivation $\boldsymbol{\mathcal{D}}(\mathcal{C})$ with conclusion $\tau = \sigma$ that has precisely the classes $[\tau = \sigma]^{u_i}$ (for $i \in 1, \ldots, n$) of open marked assumptions. Thereby $\boldsymbol{\mathcal{D}}(\mathcal{C})$ contains at least one application of ARROW or ARROW/FIX if and only if $\mathcal{C}$ contains at least one branching DECOMP.

This statement can be shown by induction on the depth $|\mathcal{C}|$ of $\mathcal{C}$.

The base case of the induction is obvious (analogously to the base case of the proof by induction in (a)).

For the induction step we consider only the case of a partial consistency-unfolding $\mathcal{C}$ for which the inductive definition of $\boldsymbol{\mathcal{D}}(\mathcal{C})$ in Definition 6.4.3 in the outermost induction step needs an application of the fifth clause in Figure 6.5 (the argumentation in the other four cases to consider is either much easier or, in the case of a needed application of the third clause in Figure 6.5, involves a very similar argumentation as in the case spelled out for the inductive proof in (a)). That is, we consider a partial consistency-unfolding $\mathcal{C}$ of the form

$$\begin{array}{cc}
\multicolumn{2}{c}{(\tilde{\tau}_1 \to \tilde{\tau}_2 = \tilde{\sigma}_1 \to \tilde{\sigma}_2)^u} \\
\hdashline
(\tilde{\tau}_1 = \tilde{\sigma}_1)^{\boldsymbol{m}_1} & (\tilde{\tau}_2 = \tilde{\sigma}_2)^{\boldsymbol{m}_2} \\
\mathcal{C}_1 & \mathcal{C}_2 \\
[\tilde{\tau}_1 \to \tilde{\tau}_2 = \tilde{\sigma}_1 \to \tilde{\sigma}_2]^u \quad & \quad [\tilde{\tau}_1 \to \tilde{\tau}_2 = \tilde{\sigma}_1 \to \tilde{\sigma}_2]^u \\
\quad \big\{[\tau_i = \sigma_i]^{u_i}\big\}_{i=1,\ldots,n} & \quad \big\{[\tau_i = \sigma_i]^{u_i}\big\}_{i=1,\ldots,n}
\end{array}
\ \text{DECOMP}$$

for some $\tilde{\tau}_1, \tilde{\tau}_2, \tilde{\sigma}_1, \tilde{\sigma}_2 \in \mu Tp$ and an assumption-marker u, where the classes of unbound leaf-occurrences of marked formulas are precisely those that belong to the displayed family $\{[\tau_i = \sigma_i]^{u_i}]_{i=1,\ldots,n}$ for some $n \in \omega$ and $\tau_i, \sigma_i \in \mu Tp$, a marker u_i for $i = 1, \ldots, n$; the respective parts in $\mathcal{C}_1$ and in $\mathcal{C}_2$ of leaf-occurrences of marked formulas that belong to one of these classes $[\tau_i = \sigma_i]^{u_i}$ are gathered at the bottom of $\mathcal{C}_1$ and of $\mathcal{C}_2$ by respective families each denoted there by $\{[\tau_i = \sigma_i]^{u_i}\}_{i=1,\ldots,n}$. Furthermore $[\tilde{\tau}_1 \to \tilde{\tau}_2 = \tilde{\sigma}_1 \to \tilde{\sigma}_2]^u$ at the bottom of $\mathcal{C}_1$ and $\mathcal{C}_2$ denotes the class of all such leaf-occurrences of $(\tilde{\tau}_1 \to \tilde{\tau}_2 = \tilde{\sigma}_1 \to \tilde{\sigma}_2)^u$ in $\mathcal{C}_1$ and respectively in $\mathcal{C}_2$ that are not bound back in $\mathcal{C}_1$ or in $\mathcal{C}_2$, but that are bound back in $\mathcal{C}$ to its root. Since $\mathcal{C}$ is, as a partial consistency-unfolding, also a f.t.o.c. with b.l.o.m.f., we conclude that there is at least one such leaf-occurrence of $(\tilde{\tau}_1 \to \tilde{\tau}_2 = \tilde{\sigma}_1 \to \tilde{\sigma}_2)^u$ in either the part $\mathcal{C}_1$ or in the part $\mathcal{C}_2$ of $\mathcal{C}$ that is bound back to the root of $\mathcal{C}$. By the induction hypothesis we now find that $\boldsymbol{\mathcal{D}}(\mathcal{C}_1)$ and $\boldsymbol{\mathcal{D}}(\mathcal{C}_2)$ are **e-HB**$_{\mathbf{0}}^{=}$-derivations with respective conclusions $\tilde{\tau}_1 = \tilde{\sigma}_1$ and $\tilde{\tau}_2 = \tilde{\sigma}_2$, which together possess precisely the classes $[\tilde{\tau}_1 \to \tilde{\tau}_2 = \tilde{\sigma}_1 \to \tilde{\sigma}_2]^u$ and $[\tau_i = \sigma_i]^{u_i}$ for $i \in \{1, \ldots, n\}$ as undischarged assumption classes that are inhabited in either $\boldsymbol{\mathcal{D}}(\mathcal{C}_1)$ or $\boldsymbol{\mathcal{D}}(\mathcal{C}_2)$. From this it follows that the result $\boldsymbol{\mathcal{D}}(\mathcal{C})$

$$\dfrac{\begin{array}{c}[\tilde{\tau}_1 \to \tilde{\tau}_2 = \tilde{\sigma}_1 \to \tilde{\sigma}_2]^u \\ \big\{[\tau_i = \sigma_i]^{u_i}\big\}_{i=1,\ldots,n} \\ \boldsymbol{\mathcal{D}}(\mathcal{C}_1) \\ \tilde{\tau}_1 = \tilde{\sigma}_1\end{array} \qquad \begin{array}{c}[\tilde{\tau}_1 \to \tilde{\tau}_2 = \tilde{\sigma}_1 \to \tilde{\sigma}_2]^u \\ \big\{[\tau_i = \sigma_i]^{u_i}\big\}_{i=1,\ldots,n} \\ \boldsymbol{\mathcal{D}}(\mathcal{C}_2) \\ \tilde{\tau}_2 = \tilde{\sigma}_2\end{array}}{\tilde{\tau}_1 \to \tilde{\tau}_2 = \tilde{\sigma}_1 \to \tilde{\sigma}_2}\ \text{ARROW/FIX}, u$$

of applying the reflection function $\boldsymbol{\mathcal{D}}$ to $\mathcal{C}$ is an $\mathbf{e}$-$\mathbf{HB}_{\mathbf{0}}^{=}$-derivation with conclusion $\tilde{\tau}_1 \to \tilde{\tau}_2 = \tilde{\sigma}_1 \to \tilde{\sigma}_2$ and whose undischarged assumption classes are precisely those of the family $\{[\tau_i = \sigma_i]^{u_i}]_{i=1,\ldots,n}$ (in particular we concluded and used here that the side-condition $\mathbf{C}$ on the application of ARROW/FIX at the bottom of $\boldsymbol{\mathcal{D}}(\mathcal{C})$ is satisfied). In this case clearly $\mathcal{C}$ contains at least one branching DECOMP as well as $\boldsymbol{\mathcal{D}}(\mathcal{C})$ contains at least one application of ARROW/FIX. Thus the statement to be shown for the induction step has been proved here.

(c) Part (iii) of the lemma is a consequence of the two more general statements

> "$\boldsymbol{\mathcal{D}} \circ \boldsymbol{\mathcal{C}}\ (\mathcal{D}) = \mathcal{D}$ holds for all derivations $\begin{array}{c}\{\,[\tau_i = \sigma_i]^{u_i}\,\}_{i=1,\ldots,n} \\ \mathcal{D} \\ \tau = \sigma\end{array}$
>
> in $\mathbf{e}$-$\mathbf{HB}_{\mathbf{0}}^{=}$ with conclusion $\tau = \sigma$ (where $n \in \omega$, and where $\tau, \sigma, \tau_1, \ldots, \tau_n, \sigma_1, \ldots, \sigma_n \in \mu Tp$) that contain precisely the classes $[\tau_i = \sigma_i]^{u_i}$ for $i = 1, \ldots, n$ of marked open assumptions."

and

> "$\boldsymbol{\mathcal{C}} \circ \boldsymbol{\mathcal{D}}\ (\mathcal{C}) = \mathcal{C}$ holds for all partial consistency-unfoldings
>
> $\begin{array}{c}(\tau = \sigma)^{\boldsymbol{m}} \\ \mathcal{C} \\ \{\,[\tau_i = \sigma_i]^{u_i}\,\}_{i=1,\ldots,n}\end{array}$ of $\tau = \sigma$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ (where $n \in \omega$ and
>
> where $\tau, \sigma, \tau_1, \ldots, \tau_n, \sigma_1, \ldots, \sigma_n \in \mu Tp$) that contain precisely the classes $[\tau_i = \sigma_i]^{u_i}$ for $i = 1, \ldots, n$ of unbound leaf-occurrences of marked formulas."

These two assertions can be shown in a straightforward way by respective inductions on $|\mathcal{D}|$ and respectively on $|\mathcal{C}|$ by distinguishing the respective 5 cases of the inductive clauses in the definition of the reflection operations $\boldsymbol{\mathcal{D}}$ and $\boldsymbol{\mathcal{C}}$ in Definition 6.4.3.

□

For all derivations $\mathcal{D}$ in $\mathbf{e}$-$\mathbf{HB}_{\mathbf{0}}^{=}$ without open assumptions and for all consistency-unfoldings $\mathcal{C}$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ that the reflection of each other, i.e. for which $\boldsymbol{\mathcal{D}}(\mathcal{C}) = \mathcal{D}$

Figure 6.7: Example consisting of a consistency-unfolding $\mathcal{C}_1$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ and of a derivation $\mathcal{D}_1$ in $\mathbf{e\text{-}HB}_{\mathbf{0}}^{=}$ without open assumptions that are reflections of each other, i.e. for which $\boldsymbol{\mathcal{D}}(\mathcal{C}_1) = \mathcal{D}_1$ and $\boldsymbol{\mathcal{C}}(\mathcal{D}_1) = \mathcal{C}_1$ holds:

$$\mathcal{D}_1 \left\{ \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{(\tau=\sigma)^u \qquad \dfrac{}{\bot=\bot}}{\tau\to\bot=\sigma\to\bot}\,\text{ARROW}}{\tau=\sigma\to\bot}\,\text{FOLD}_l \qquad \dfrac{}{\bot=\bot}}{\tau\to\bot=(\sigma\to\bot)\to\bot}\,\text{ARROW}}{\tau\to\bot=\sigma}\,\text{FOLD}_r}{\underbrace{\mu\alpha.\,(\alpha\to\bot)}_{\equiv\tau}=\underbrace{\mu\beta.\,((\beta\to\bot)\to\bot)}_{\equiv\sigma}}\,\text{FOLD}_l/\text{FIX},\,u \right.$$

$$\mathcal{C}_1 \left\{ \begin{array}{cl}
\big(\mu\alpha.\,(\alpha\to\bot)=\mu\beta.\,((\beta\to\bot)\to\bot)\big)^u & \\ \hline
\tau\to\bot=\sigma & \text{UNFOLD}_l \\ \hline
\tau\to\bot=(\sigma\to\bot)\to\bot & \text{UNFOLD}_r \\ \hdashline
\begin{array}{cl} \tau=\sigma\to\bot & \\ \hline \tau\to\bot=\sigma\to\bot & \text{UNFOLD}_l \\ \hdashline (\tau=\sigma)^u \qquad \bot=\bot & \text{DECOMP} \end{array} \qquad \bot=\bot & \text{DECOMP}
\end{array} \right.$$

and $\boldsymbol{\mathcal{C}}(\mathcal{D}) = \mathcal{C}$ holds, we say that $\mathcal{D}$ and $\mathcal{C}$ are *dual* (*to each other*), or more explicitly, that $\mathcal{C}$ is the *dual consistency-unfolding in* $\mathbf{AK}_{\mathbf{0}}^{=}$ *for* $\mathcal{D}$, and that $\mathcal{D}$ is the *dual derivation in* $\mathbf{e\text{-}HB}_{\mathbf{0}}^{=}$ *for* $\mathcal{C}$.

Now we give an example consisting of a consistency-unfolding in $\mathbf{AK}_{\mathbf{0}}^{=}$ and of a derivation in $\mathbf{e\text{-}HB}_{\mathbf{0}}^{=}$ that are dual to each other. For mainly typographical reasons we consider a simpler consistency-unfolding than that in Example 6.1.1, and accordingly, a simpler derivation than that of Example 6.1.2.

Example 6.5.2 (Dual $\mathbf{e\text{-}HB}_{\mathbf{0}}^{=}$-deriv. and consistency-unfolding in $\mathbf{AK}_{\mathbf{0}}^{=}$). We consider the two recursive types

$$\tau \equiv \mu\alpha.\,(\alpha\to\bot)\,, \qquad\qquad \sigma \equiv \mu\beta.\,((\beta\to\bot)\to\bot)\,,$$

which are respectively equal to the strongly equivalent recursive types τ_1 and σ_1 in[8] Example 3.6.2. Therefore $\tau =_\mu \sigma$ is the case here.

A consistency-unfolding of $\tau = \sigma$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ can be found, as this is true in general for all equations between strongly equivalent recursive types, in an algorithmic way: by 'growing' a derivation-tree of consequences in $\mathbf{AK}_{\mathbf{0}}^{=}$ from $\tau = \sigma$ in downwards

[8] We have avoided to use τ_1 and σ_1 from Example 3.6.2 directly in the desire to avoid distracting subscripts here.

direction through (reasonable[9]) stepwise extensions with applicable applications in $\mathbf{AK}_{\mathbf{0}}^{=}$ until in all leaves of the derivation-tree either axioms (REFL) or such formulas are found, which have been encountered before in the thread down from the root $\tau = \sigma$ and which therefore can be bound back to these respective earlier occurrences.

In this manner the f.t.o.c. with b.l.o.m.f. $\mathcal{C}_1$ from $\tau = \sigma$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ in Figure 6.7 is found, which is easily recognizable as a consistency-unfolding of $\tau = \sigma$ in $\mathbf{AK}_{\mathbf{0}}^{=}$. Its reflection $\boldsymbol{\mathcal{D}}(\mathcal{C}_1)$ can be seen to be the derivation $\mathcal{D}_1$ in $\mathbf{e\text{-}HB}_{\mathbf{0}}^{=}$ also depicted in Figure 6.7, i.e. $\boldsymbol{\mathcal{D}}(\mathcal{C}_1) = \mathcal{D}_1$. Vice versa, also $\boldsymbol{\mathcal{C}}(\mathcal{D}_1) = \mathcal{C}_1$ can be checked easily. Hence $\mathcal{C}_1$ and $\mathcal{D}_1$ are dual to each other. ■

It is certainly not difficult to find the dual $\mathbf{e\text{-}HB}_{\mathbf{0}}^{=}$-derivation $\boldsymbol{\mathcal{D}}(\mathcal{C})$ for the consistency-unfolding $\mathcal{C}$ in Figure 6.1 as well as the dual consistency-unfolding $\boldsymbol{\mathcal{C}}(\mathcal{D})$ for the derivation $\mathcal{D}$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ in Figure 6.4. And it is easy to convince oneself, as we have informally recognized already in Section 6.1, that $\mathcal{C}$ and $\mathcal{D}$ are *not* reflections of each other and hence are not dual to each other.

In the light of the example above it is easy to explain why, with the aim of establishing a satisfying duality result, we have chosen to extend the system $\mathbf{HB}_{\mathbf{0}}^{=}$ first by some additional inference rules to the system $\mathbf{e\text{-}HB}_{\mathbf{0}}^{=}$ before only later defining mutual transformations between partial consistency-unfoldings in $\mathbf{AK}_{\mathbf{0}}^{=}$ and derivations in the extended system $\mathbf{e\text{-}HB}_{\mathbf{0}}^{=}$. If we had not done so, then we would not have been able to discharge the open marked assumption $(\tau = \sigma)^u$ in a derivation Refl$(\mathcal{C}_1)$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ that arises by plain reflection from the consistency-unfolding $\mathcal{C}_1$ from Figure 6.7. Similarly as described for the example considered in Section 6.1, we would have had to enlarge Refl$(\mathcal{C}_1)$ above this marked assumption by two additional applications (one of FOLD$_l$ and one of FOLD$_r$) before being able to discharge the newly occurring open marked assumption $(\tau \to \bot = (\sigma \to \bot) \to \bot)^u$ in a thereby created derivation Refl$(\mathcal{C}_1)^*$ at an application of ARROW/FIX that results by renaming from the bottommost application of ARROW in Refl$(\mathcal{C}_1)$.[10] But due to the presence of the rules FOLD$_{l/r}$/FIX in $\mathbf{e\text{-}HB}_{\mathbf{0}}^{=}$ it was possible here to transform the plain reflection Refl$(\mathcal{C}_1)$ of $\mathcal{C}_1$ into the derivation $\mathcal{D}_1$ in Figure 6.7 by merely renaming the bottommost application of FOLD$_l$ in Refl$(\mathcal{C}_1)$ into FOLD$_l$/FIX and by discharging the open marked assumption $(\tau = \sigma)^u$ at this application.

This look at the example from Figure 6.7 can make it clear why it is actually not possible to find a bijective and equally immediate correspondence as stated in Theorem 6.5.1 between *arbitrary* consistency-unfoldings in $\mathbf{AK}_{\mathbf{0}}^{=}$ and derivations in $\mathbf{HB}_{\mathbf{0}}^{=}$.

[9] By “reasonable” extensions of a derivation-tree through rule applications we mean here more precisely that applications of REN are only allowed as auxiliary steps for enabling applications of UNFOLD$_{l/r}$, and that extensions by applications of the rules $(\mu{-}\bot)^{\mathrm{der}\bot}_{l/r}$ always take precedence over extensions with any other rule of $\mathbf{AK}_{\mathbf{0}}^{=}$.

[10] The derivation Refl$(\mathcal{C}_1)^*$ described here is actually equal to the derivation $\mathcal{D}_2$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ depicted in Figure 6.8; strictly speaking, Refl$(\mathcal{C}_1)^*$ is equal to one of the four $\mathbf{HB}_{\mathbf{0}}^{=}$-derivations that are denoted by the prooftree $\mathcal{D}_2$ shown in Figure 6.8 since for the two pairs of successive FOLD$_{l/r}$-applications the order in which these two applications appear after each other has not been fixed there.

6.6 Specializing the Duality to Derivations in $\mathbf{HB}_{\mathbf{0}}^{=}$

The duality theorem, Theorem 6.5.1, leaves open the question how the particular class of those consistency-unfoldings in $\mathbf{AK}_{\mathbf{0}}^{=}$ that are the images under the reflection function $\boldsymbol{\mathcal{C}}$ of derivations in the basic system $\mathbf{HB}_{\mathbf{0}}^{=}$ can formally be characterized. Closer examination in this section is able to show that such consistency-unfoldings have a particular property, which we call "property **D**" in the definition below. With this notion we will then be able to formulate and prove a specialized version of Theorem 6.5.1.

Definition 6.6.1 (The property D for f.t.o.c.'s with b.l.o.m.f.'s). Let τ and σ be recursive types. A f.t.o.c. with b.l.o.m.f.'s $\mathcal{C}$ of $\tau = \sigma$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ *satisfies the property* **D** iff all bound leaf-occurrences of marked formulas $(\rho = \chi)^u$ in $\mathcal{C}$ (with arbitrary $\rho, \chi \in \mu Tp$ and markers u) are bound back to respective premises of branchings DECOMP. ⊠

Example 6.6.2. It is easy to see that the consistency-unfolding in $\mathbf{AK}_{\mathbf{0}}^{=}$ in Figure 6.4 satisfies the property **D**, whereas both the consistency-unfoldings $\mathcal{C}$ in Figure 6.1 and $\mathcal{C}_1$ in Figure 6.7 do not. ■

We denote by $pCU^{\mathbf{D}}(\mathbf{AK}_{\mathbf{0}}^{=})$ the set of partial consistency-unfoldings in $\mathbf{AK}_{\mathbf{0}}^{=}$ with the property **D**; and by $CU^{\mathbf{D}}(\mathbf{AK}_{\mathbf{0}}^{=})$ the set of consistency-unfoldings in $\mathbf{AK}_{\mathbf{0}}^{=}$ with the property **D**.

Generally, the fulfilledness of the property **D** for f.t.o.c. with b.l.o.m.f.'s can be illustrated as follows. Let $\mathcal{C}$ be a f.t.o.c. with b.l.o.m.f. from $\tau = \sigma$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ that satisfies the property **D**. Then, with respect to an arbitrary back-bound leaf-occurrence of a marked formula $(\rho = \chi)^u$ in $\mathcal{C}$ that is displayed as the single leaf in the representation of $\mathcal{C}$ on the left below, $\mathcal{C}$ can, *for example*, be written as

$$\begin{array}{c}\tau=\sigma\\ \mathcal{C}\\ (\rho=\chi)^u\end{array}\quad\left\{\quad\begin{array}{cc}\multicolumn{2}{c}{\tau=\sigma}\\ \multicolumn{2}{c}{\mathcal{C}_1}\\ \multicolumn{2}{c}{(\rho_1\to\rho_2=\chi_1\to\chi_2)^u}\\ \hdashline \rho_1=\chi_1 & \rho_2=\chi_2\\ \mathcal{C}_{1a} & \mathcal{C}_{2a}\\ (\rho_1\to\rho_2=\chi_1\to\chi_2)^u & \end{array}\ \text{DECOMP}\right. \tag{6.27}$$

for some $\rho_1, \rho_2, \chi_1, \chi_2 \in \mu Tp$ such that $\rho \equiv \rho_1 \to \rho_2$ and $\chi \equiv \chi_1 \to \chi_2$ and with some f.t.o.c.'s with m.f. $\mathcal{C}_1$ from $\tau = \sigma$, $\mathcal{C}_{1a}$ from $\rho_1 = \chi_1$ and $\mathcal{C}_{2a}$ from $\rho_2 = \chi_2$ in $\mathbf{AK}_{\mathbf{0}}^{=}$, where the displayed leaf-occurrence of $(\rho = \chi)^u$ on the left side is the leaf-occurrence of $(\rho_1 \to \rho_2 = \chi_1 \to \chi_2)^u$ on the right side and where the two occurrences of the marked formula $(\rho_1 \to \rho_2 = \chi_1 \to \chi_2)^u$ in the representation of $\mathcal{C}$ on the right side are assumed to be linked by a backbinding. Hence the considered leaf-occurrence of $(\rho = \chi)^u$ displayed on the right in (6.27) is indeed bound back to the upper premise of a branching DECOMP in $\mathcal{C}$ (as this is demanded by the condition **D** on $\mathcal{C}$). And the words "*for example*" above are intended to convey that the leaf-occurrence of $(\rho = \chi)^u$ displayed in $\mathcal{C}$ on the left in (6.27), which corresponds to the leaf-occurrence of $(\rho_1 \to \rho_2 = \chi_1 \to \chi_2)^u$ displayed at the bottom of $\mathcal{C}_{1a}$ on the right, could also occur at the bottom of $\mathcal{C}_{2a}$ in a similar representation of $\mathcal{C}$ as on the right hand side of (6.27).

Now we are able to formulate the following specialized version of Theorem 6.5.1. It stated that there does also exist a *duality* via the reflection functions $\boldsymbol{\mathcal{C}}$ and $\boldsymbol{\mathcal{D}}$ between derivations without open assumptions in the basic, not extended, system $\mathbf{HB}_{\mathbf{0}}^{=}$ and consistency-unfoldings in $\mathbf{AK}_{\mathbf{0}}^{=}$ with the property $\mathbf{D}$.

Theorem 6.6.3. (Specializing the duality in Theorem 6.5.1: A duality between between derivations in $\mathbf{HB}_{\mathbf{0}}^{=}$ and consistency-unfoldings in $\mathbf{AK}_{\mathbf{0}}^{=}$ with the property D) *The reflection functions $\boldsymbol{\mathcal{D}}$ and $\boldsymbol{\mathcal{C}}$ defined in Definition 6.4.3 yield* also *a bijective functional relationship between derivations in $\mathbf{HB}_{\mathbf{0}}^{=}$ without open assumption classes and consistency-unfoldings in $\mathbf{AK}_{\mathbf{0}}^{=}$ with the property $\mathbf{D}$ in the sense as expressed by the following three statements:*

(i) For every derivation $\mathcal{D}$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ without open assumption classes and with conclusion $\tau = \sigma$ its reflection $\boldsymbol{\mathcal{C}}(\mathcal{D})$ is a consistency-unfolding in $\mathbf{AK}_{\mathbf{0}}^{=}$ of $\tau = \sigma$ with the property $\mathbf{D}$.

(ii) For every consistency-unfolding $\mathcal{C}$ of $\tau = \sigma$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ with the property $\mathbf{D}$ its reflection $\boldsymbol{\mathcal{D}}(\mathcal{C})$ is a derivation in $\mathbf{HB}_{\mathbf{0}}^{=}$ with conclusion $\tau = \sigma$ and without open assumption classes.

(iii) The restrictions

$$\begin{aligned} \boldsymbol{\mathcal{D}}|_{\mathcal{D}er_{\emptyset}(\mathbf{HB}_{\mathbf{0}}^{=})} &: \mathcal{D}er_{\emptyset}(\mathbf{HB}_{\mathbf{0}}^{=}) \longrightarrow \mathcal{CU}^{\mathbf{D}}(\mathbf{AK}_{\mathbf{0}}^{=}), \quad \textit{and} \\ \boldsymbol{\mathcal{C}}|_{\mathcal{CU}^{\mathbf{D}}(\mathbf{AK}_{\mathbf{0}}^{=})} &: \mathcal{CU}^{\mathbf{D}}(\mathbf{AK}_{\mathbf{0}}^{=}) \longrightarrow \mathcal{D}er_{\emptyset}(\mathbf{HB}_{\mathbf{0}}^{=}) \end{aligned}$$

of the reflection functions $\boldsymbol{\mathcal{D}}$ and $\boldsymbol{\mathcal{C}}$ to, respectively, the set $\mathcal{D}er_{\emptyset}(\mathbf{HB}_{\mathbf{0}}^{=})$ of derivations in $\mathbf{AK}_{\mathbf{0}}^{=}$ without open assumptions, and to the set $\mathcal{CU}^{\mathbf{D}}(\mathbf{AK}_{\mathbf{0}}^{=})$ of consistency-unfoldings in $\mathbf{AK}_{\mathbf{0}}^{=}$ with the property $\mathbf{D}$ are inverses of each other.

Proof. The proof of this theorem is a close-grained version of the proof of Theorem 6.5.1. In particular, for the items (i) and (ii) in the theorem it has to be shown more generally that (1) the reflection function $\boldsymbol{\mathcal{C}}$ maps derivations $\mathcal{D}$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ to partial consistency-unfoldings $\mathcal{C} = \boldsymbol{\mathcal{C}}(\mathcal{D})$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ with the property $\mathbf{D}$, and respectively, that reversely (2) the reflection function $\boldsymbol{\mathcal{D}}$ reversely maps partial consistency-unfoldings $\mathcal{C}$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ with the property $\mathbf{D}$ to derivations $\mathcal{D} = \boldsymbol{\mathcal{D}}(\mathcal{C})$ in $\mathbf{HB}_{\mathbf{0}}^{=}$. These two assertions are illustrated with more detail by the following picture (where $\tau, \sigma \in \mu Tp$, $n \in \omega$ and $\tau_i, \sigma_i \in \mu Tp$ as well as u_i a marker for $1 \leq i \leq n$):

$$\boxed{\begin{array}{l} \{\, [\tau_i = \sigma_i]^{u_i} \,\}_{i=1,\ldots,n} \\ \qquad \mathcal{D} \\ \quad \tau = \sigma \end{array}} \quad \begin{array}{c} \xmapsto{\boldsymbol{\mathcal{C}}} \\ \xleftarrow[\boldsymbol{\mathcal{D}}]{} \end{array} \quad \boxed{\begin{array}{l} \quad (\tau = \sigma)^{\boldsymbol{m}} \\ \qquad \mathcal{C} \\ \{\, [\tau_i = \sigma_i]^{u_i} \,\}_{i=1,\ldots,n} \end{array}}$$

derivation of $\tau = \sigma$ in the system $\mathbf{HB}_{\mathbf{0}}^{=}$ with the open assumption classes $[\tau_i = \sigma_i]^{u_i}$ $(1 \leq i \leq n)$

partial consistency-unfolding of $\tau = \sigma$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ *satisfying the property* **D** and with the classes $[\tau_i = \sigma_i]^{u_i}$ of unbound leaf-occurrences of marked formulas $(1 \leq i \leq n)$

The assertion (1) can be shown for all $\mathbf{HB}_{\mathbf{0}}^{=}$-derivations $\mathcal{D}$ by induction on $|\mathcal{D}|$. In such a proof the following fact is exploited: for $\mathbf{HB}_{\mathbf{0}}^{=}$-derivations $\mathcal{D}$, the third inductive clause in Figure 6.6 regarding rules $R^{(d)}$/FIX with $R^{(d)} \in \{\text{FOLD}_{l/r}, \text{REN}\}$ is never used in the inductive definition of $\boldsymbol{\mathcal{C}}(\mathcal{D})$ since these rules are not part of the system $\mathbf{HB}_{\mathbf{0}}^{=}$. Hence no upper premise of a rule application (different from a DECOMP-branching) in $\boldsymbol{\mathcal{C}}(\mathcal{D})$ can be a marked formula, and therefore $\boldsymbol{\mathcal{C}}(\mathcal{D})$ ultimately fulfills the property **D** for all $\mathbf{HB}_{\mathbf{0}}^{=}$-derivations $\mathcal{D}$.

And statement (2) can be shown for all partial consistency-unfoldings $\mathcal{C}$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ with the property **D** by induction on $|\mathcal{C}|$. For this proof the following observation is essential: for arbitrary partial consistency-unfoldings $\mathcal{C}$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ with the property **D**, in the inductive definition of $\boldsymbol{\mathcal{D}}(\mathcal{C})$ the third clause in Figure 6.5 (regarding f.t.o.c.'s with b.l.o.m.f. with a marked formula at their root-position that is followed by an application of a rule $R^{(cu)} \in \{\text{UNFOLD}_{l/r}, \text{REN}\}$) is never used (since due to the property **D** on $\mathcal{C}$ no leaf is bound back in $\mathcal{C}$ to the upper premise of such a rule and hence the premise of a rule $\text{UNFOLD}_{l/r}$ or REN must always be an unmarked formula). Therefore it follows that, for all consistency-unfoldings $\mathcal{C}$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ with the property **D**, the reflection $\boldsymbol{\mathcal{D}}(\mathcal{C})$ of $\mathcal{D}$ does not contain applications of rules $\text{FOLD}_{l/r}$/FIX and REN/FIX and hence must be a derivation in $\mathbf{HB}_{\mathbf{0}}^{=}$.

The item (iii) of the theorem, the fact, that the restriction of the reflection function $\boldsymbol{\mathcal{C}}$ to $\mathbf{HB}_{\mathbf{0}}^{=}$-derivations without open assumption classes and the restriction of the reflection-operation $\boldsymbol{\mathcal{D}}$ to consistency-unfoldings with the property **D** are inverses of each other, is an immediate consequence of item (iii) of Theorem 6.5.1 and of the items (i) and (ii) of the theorem here.

□

We can illustrate this theorem with a consistency-unfolding in $\mathbf{AK}_{\mathbf{0}}^{=}$ and its dual derivation in $\mathbf{HB}_{\mathbf{0}}^{=}$, which are closely related to the consistency-unfolding and the derivation given in Example 6.5.2.

Example 6.6.4 (Dual $\mathbf{HB}_{\mathbf{0}}^{=}$-derivation and consistency-unfolding in $\mathbf{AK}_{\mathbf{0}}^{=}$ with the property D). Let τ and σ be the recursive types $\tau \equiv \mu\alpha.\,(\alpha \to \bot)$ and $\sigma \equiv \mu\beta.\,((\beta \to \bot) \to \bot)$ as in Example 6.5.2, which are strongly equivalent (as

Figure 6.8: Example consisting of a consistency-unfolding $\mathcal{C}_2$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ with the property **D** and of a derivation $\mathcal{D}_2$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ without open assumptions that are reflections of each other, i.e. for which $\boldsymbol{\mathcal{D}}(\mathcal{C}_2) = \mathcal{D}_2$ and $\boldsymbol{\mathcal{C}}(\mathcal{D}_2) = \mathcal{C}_2$ holds:

$$\mathcal{D}_2 \left\{ \cfrac{\cfrac{\cfrac{\cfrac{\cfrac{(\tau \to \bot = (\sigma \to \bot) \to \bot)^u}{\tau = \sigma} \qquad \cfrac{}{\bot = \bot}}{\tau \to \bot = \sigma \to \bot}\ \text{ARROW}}{\tau = \sigma \to \bot}\ \text{FOLD}_l \qquad \cfrac{}{\bot = \bot}}{\tau \to \bot = (\sigma \to \bot) \to \bot}\ \text{ARR./FIX}, u}{\underbrace{\mu\alpha.\,(\alpha \to \bot)}_{\equiv \tau} = \underbrace{\mu\beta.\,((\beta \to \bot) \to \bot)}_{\equiv \sigma}}\ \text{FOLD}_{l/r} \right.$$

$$\mathcal{C}_2 \left\{ \begin{array}{c} \mu\alpha.\,(\alpha \to \bot) = \mu\beta.\,((\beta \to \bot) \to \bot) \\ \hline\hline (\tau \to \bot = (\sigma \to \bot) \to \bot)^u \quad \text{UNFOLD}_{l/r} \\ \hdashline \tau = \sigma \to \bot \qquad\qquad \bot = \bot \quad \text{DECOMP} \\ \hline \tau \to \bot = \sigma \to \bot \quad \text{UNFOLD}_l \\ \hdashline \tau = \sigma \qquad \bot = \bot \quad \text{DECOMP} \\ \hline\hline (\tau \to \bot = (\sigma \to \bot) \to \bot)^u \end{array} \right.$$

we already noticed in Example 3.6.2).

As this is possible in general with respect to two arbitrary strongly equivalent recursive types, a formal proof for $\tau = \sigma$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ can be found in an algorithmic way: by developing a prooftree in upwards-direction by (reasonable[11]) extensions with applications of applicable rules of $\mathbf{HB}_{\mathbf{0}}^{=}$ until either axioms (REFL) are met or such formulas that have been encountered before in the thread from the root of the prooftree upwards as the conclusions of applications of ARROW or ARROW/FIX (in this case such formulas at the top of a found derivation are made into open assumptions that are subsequently bound to the respective application that in its turn is converted into an application of ARROW/FIX). In such a manner the shortest possible[12] derivation of $\tau = \sigma$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ is actually of the form $\mathcal{D}_2$ depicted in Figure 6.8.

It can be seen that $\boldsymbol{\mathcal{C}}(\mathcal{D}_2) = \mathcal{C}_2$ holds, i.e. that the reflection $\boldsymbol{\mathcal{C}}(\mathcal{D}_2)$ of $\mathcal{D}_2$ is the

[11] Hereby we mean that extensions by REN-applications are only introduced in situations where they are needed to facilitate applications of $\text{FOLD}_{l/r}$, and that extensions by rules $(\mu - \bot)^{\bot\text{der}}_{l/r}$ always take precedence over extensions with any other rule of $\mathbf{HB}_{\mathbf{0}}^{=}$.

[12] Since there appear two pairs of respectively interchangeable successive applications of FOLD_l and FOLD_r in each such derivation, there are actually four derivations of $\tau = \sigma$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ of smallest depth (namely of depth 7). Nevertheless it seems reasonable to speak of *the* shortest derivation $\mathcal{D}_2$ of $\tau = \sigma$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ with a form as illustrated by the symbolic prooftree in Figure 6.8.

consistency-unfolding $\mathcal{C}_2$ of $\tau = \sigma$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ also shown in Figure 6.8. Vice versa also $\boldsymbol{\mathcal{D}}(\mathcal{C}_2) = \mathcal{D}_2$ can quickly be verified for the reflection of the consistency-unfolding $\mathcal{C}_2$. Since the only back-bound leaf-occurrence of a marked formula in $\mathcal{C}_2$ is bound back to the upper premise of a branching DECOMP, $\mathcal{C}_2$ fulfills the property $\mathbf{D}$.

Hence the $\mathbf{HB}_{\mathbf{0}}^{=}$-derivation $\mathcal{D}_2$ without open assumptions and the consistency-unfolding $\mathcal{C}_2$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ with the property $\mathbf{D}$, which are both shown in Figure 6.8, are dual to each other. ■

6.7 Concluding Remarks and a Consequence of the Duality

We want to close this chapter with a remark about a 'hierarchy' of dualities that can be extracted from our theorems and proofs here, and by an application of our main duality result to give an alternative soundness proof for the variant Brandt-Henglein system $\mathbf{HB}_{\mathbf{0}}^{=}$.

Remark 6.7.1 ('Hierarchy' of dualities). Apart from the correspondences stated by the duality theorems, Theorem 6.5.1 and Theorem 6.6.3, in the proofs we have found also other, similar correspondence statements that deserve some attention on their own merit and in relation to the other statements. For instance, we have shown in the proof of Theorem 6.5.1 that the reflection functions $\boldsymbol{\mathcal{D}}$ and $\boldsymbol{\mathcal{C}}$ are also bijective between the set $pCU(\mathbf{AK}_{\mathbf{0}}^{=})$ of partial consistency-unfoldings in $\mathbf{AK}_{\mathbf{0}}^{=}$ and the set $\mathcal{D}er(\textbf{e-HB}_{\mathbf{0}}^{=})$ of derivations in $\textbf{e-HB}_{\mathbf{0}}^{=}$ with (possibly) open assumptions. And in the proof of Theorem 6.6.3 we found that $\boldsymbol{\mathcal{D}}$ and $\boldsymbol{\mathcal{C}}$ are bijective between the set $pCU^{\mathbf{D}}(\mathbf{AK}_{\mathbf{0}}^{=})$ of partial consistency-unfoldings in $\mathbf{AK}_{\mathbf{0}}^{=}$ with the property $\mathbf{D}$ and the set $\mathcal{D}er(\mathbf{HB}_{\mathbf{0}}^{=})$ of derivations in $\mathbf{HB}_{\mathbf{0}}^{=}$ with (possibly) open assumptions.

For giving an overview of the different kinds of duality statements between $\mathbf{AK}_{\mathbf{0}}^{=}$ and $\mathbf{HB}_{\mathbf{0}}^{=}$ that have been reached, we have gathered five of them in the pictures (6.28)–(6.32) shown in Figure 6.9. Hereby (6.29) and (6.30) illustrate the additional duality statements mentioned first, and respectively second, above. (6.31) and (6.32) stand for the assertions of Theorem 6.5.1 and Theorem 6.6.3, respectively. And furthermore, a duality statement not mentioned previously is indicated in (6.28): the reflection functions $\boldsymbol{\mathcal{C}}$ and $\boldsymbol{\mathcal{D}}$ are also bijective on the domains on which they have been defined; this can be shown analogously to the proof of Theorem 6.5.1, (iii). This means that there is also a duality between pseudo-derivations in $\textbf{e-HB}_{\mathbf{0}}^{=}$ and f.t.o.c.'s with b.l.o.m.f. in $\mathbf{AK}_{\mathbf{0}}^{=}$.

In the figure at the bottom of Figure 6.9 we have furthermore indicated how the particular duality statements can be reached from the 'basic duality' (6.28) by stepwise restriction of the domain of the reflection functions $\boldsymbol{\mathcal{C}}$ and $\boldsymbol{\mathcal{D}}$.

Theorem 6.5.1, enables us to carry out the following alternative proof for the soundness part in Theorem 5.1.20, in which the soundness of $\mathbf{HB}_{\mathbf{0}}^{=}$ with respect to $=_{\mu}$ is 'reduced' to the fact that $\mathbf{AK}_{\mathbf{0}}^{=}$-consistency implies strong equivalence (according to Theorem 5.2.13).

Figure 6.9: The following five kinds of duality statements are induced by the reflection functions $\boldsymbol{\mathcal{C}}$ and $\boldsymbol{\mathcal{D}}$. A figure below shows how these can be ordered according to their 'degree of specialization'.

$$\boxed{\begin{array}{l}\text{pseudo-derivation } \mathcal{D} \\ \text{of } \tau=\sigma \text{ in } \mathbf{e\text{-}HB_0^=}\end{array}} \quad \underset{\boldsymbol{\mathcal{D}}}{\overset{\boldsymbol{\mathcal{C}}}{\substack{\longmapsto \\ \longmapsfrom}}} \quad \boxed{\begin{array}{l}\text{f.t.o.c. with b.l.o.m.f. } \mathcal{C} \\ \text{from } \tau=\sigma \text{ in } \mathbf{AK_0^=}\end{array}} \tag{6.28}$$

$$\boxed{\begin{array}{l}\text{derivation } \mathcal{D} \text{ of } \tau=\sigma \\ \text{in } \mathbf{e\text{-}HB_0^=}\end{array}} \quad \underset{\boldsymbol{\mathcal{D}}}{\overset{\boldsymbol{\mathcal{C}}}{\substack{\longmapsto \\ \longmapsfrom}}} \quad \boxed{\begin{array}{l}\text{partial consistency-unfolding } \mathcal{C} \\ \text{of } \tau=\sigma \text{ in } \mathbf{AK_0^=}\end{array}} \tag{6.29}$$

$$\boxed{\begin{array}{l}\text{derivation } \mathcal{D} \text{ of } \tau=\sigma \\ \text{in } \mathbf{HB_0^=}\end{array}} \quad \underset{\boldsymbol{\mathcal{D}}}{\overset{\boldsymbol{\mathcal{C}}}{\substack{\longmapsto \\ \longmapsfrom}}} \quad \boxed{\begin{array}{l}\text{partial consistency-unfolding } \mathcal{C} \\ \text{of } \tau=\sigma \text{ in } \mathbf{AK_0^=} \text{ with the} \\ \text{property } \mathbf{D}\end{array}} \tag{6.30}$$

$$\boxed{\begin{array}{l}\text{derivation } \mathcal{D} \text{ of } \tau=\sigma \\ \text{in } \mathbf{e\text{-}HB_0^=} \text{ without} \\ \text{open assumptions}\end{array}} \quad \underset{\boldsymbol{\mathcal{D}}}{\overset{\boldsymbol{\mathcal{C}}}{\substack{\longmapsto \\ \longmapsfrom}}} \quad \boxed{\begin{array}{l}\text{consistency-unfolding } \mathcal{C} \text{ of} \\ \tau=\sigma \text{ in } \mathbf{AK_0^=}\end{array}} \tag{6.31}$$

$$\boxed{\begin{array}{l}\text{derivation } \mathcal{D} \text{ of } \tau=\sigma \\ \text{in } \mathbf{HB_0^=} \text{ without} \\ \text{open assumptions}\end{array}} \quad \underset{\boldsymbol{\mathcal{D}}}{\overset{\boldsymbol{\mathcal{C}}}{\substack{\longmapsto \\ \longmapsfrom}}} \quad \boxed{\begin{array}{l}\text{consistency-unfolding } \mathcal{C} \text{ of} \\ \tau=\sigma \text{ in } \mathbf{AK_0^=} \text{ with the} \\ \text{property } \mathbf{D}\end{array}} \tag{6.32}$$

These five statements can be ordered, according to their 'degree of specialization', as shown by the figure below. An arrow $A \longrightarrow B$ between duality statements A and B means that statement B results from statement A by restricting the domains of the reflection functions $\boldsymbol{\mathcal{C}}$ and $\boldsymbol{\mathcal{D}}$ to the set of objects appearing on the left-hand side, and respectively, to the set of objects on the right-hand side of statement B.

$$\begin{tikzcd}
 & & (6.28) \arrow[dl] \\
 & (6.29) \arrow[dl] \arrow[dr] & \\
(6.30) \arrow[dr] & & (6.31) \arrow[dl] \\
 & (6.32) &
\end{tikzcd}$$

Alternative soundness proof for $\mathbf{HB}_{\mathbf{0}}^{=}$ *with respect to* $=_{\mu}$. Suppose that $\tau = \sigma$ is a theorem of $\mathbf{HB}_{\mathbf{0}}^{=}$, with some $\tau, \sigma \in \mu Tp$. This means that there exists a derivation $\mathcal{D}$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ with conclusion $\tau = \sigma$ and without open assumption classes; let $\mathcal{D}$ be chosen as such a derivation. Then due to Theorem 6.5.1 the reflection $\boldsymbol{\mathcal{C}}(\mathcal{D})$ of $\mathcal{D}$ is a consistency-unfolding of $\tau = \sigma$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ (which, as we remark by the way, fulfills the property $\mathbf{D}$ due to Theorem 6.6.3). Hence by Theorem 6.3.18 the equation $\tau = \sigma$ is consistent with respect to $\mathbf{AK}_{\mathbf{0}}^{=}$. And from this, Theorem 5.2.13, which states that consistency with respect to $\mathbf{AK}_{\mathbf{0}}^{=}$ entails strong equivalence, implies that τ and σ are strongly equivalent. □

Chapter 7

Transforming Derivations from $\mathbf{AC}^=$ to $\mathbf{HB}^=$

In this chapter we will develop an effective transformation from derivations in the axiomatization $\mathbf{AC}^=$ of recursive type equality by Amadio and Cardelli into derivations in the axiomatization $\mathbf{HB}^=$ given by Brandt and Henglein. More precisely, we will show that every derivation $\mathcal{D}$ in the system $\mathbf{AC}^=$ that does not contain assumptions can effectively be transformed into a derivation $\mathcal{D}'$ in the system $\mathbf{HB}^=$ of Brandt and Henglein such that $\mathcal{D}'$ has the same conclusion as $\mathcal{D}$ and $\mathcal{D}'$ does not contain open assumption classes. In Section 7.1 some later needed preparatory work will be done: in particular, the admissibility of substitution rules in $\mathbf{AC}^=$ will be proved, and it will be demonstrated that the rule μ-COMPAT of the system $\mathbf{AC}^=$ can be dispensed with in a close variant system of $\mathbf{AC}^=$. The transformation from derivations in $\mathbf{AC}^=$ into derivations in $\mathbf{HB}^=$ will then be constructed in Section 7.2, in the form of a number of lemmas that formulate the existence of constituent parts of this transformation.

7.1 Admissibility of μ-COMPAT in a Variant System of $\mathbf{AC}^=$

In this section we are concerned with gathering and investigating a couple of basic proof-theoretical properties of the proof system $\mathbf{AC}^=$ for recursive type equality due to Amadio and Cardelli. Some of these properties will be useful later, in Section 7.2, for developing a proof-theoretic transformation from derivations in $\mathbf{AC}^=$ into derivations in $\mathbf{HB}^=$.

An important part of this section is dedicated to giving a proof-theoretical demonstration of the fact that substitution is an admissible rule in the system $\mathbf{AC}^=$. More precisely, we will define three different kinds of substitution rules that act on equations between recursive types, and subsequently we will show that these rules are admissible in $\mathbf{AC}^=$. The admissibility of substitution rules of one form is in fact tacitly assumed by Amadio and Cardelli in [AmCa93] and used without proof in the completeness proof for the system $\mathbf{AC}^=$ with respect to $=_\mu$.

As a preparation for the main lemma of this section concerning the admissibility in $\mathbf{AC}^=$ of the mentioned three kinds of substitution rules, we need a couple of auxiliary technical statements. We start with a lemma about some useful reformulations of the contractiveness property $\alpha \downarrow \tau$ of a recursive type τ with respect to a type variable α, which property was defined in Definition 5.1.1.

Lemma 7.1.1 (Reformulations of the contractiveness condition "$\alpha \downarrow \tau$"). *In the items (i) and (ii) below, the condition of a recursive type τ being contractive in a type variable α is characterized, respectively, in a negative way, and then in four ways that are different from the formulation in Definition 5.1.1.*

(i) For all $\tau \in \mu Tp$ and $\alpha \in TVar$ it holds that:

$$\left.\begin{aligned} \alpha \not\downarrow \tau \iff & (\exists n \in \omega)\,(\exists\, \alpha_1, \ldots, \alpha_n \in TVar) \\ & \big[\, \tau \equiv \mu\alpha_1 \ldots \alpha_n.\, \alpha \;\;\&\;\; \alpha \not\equiv \alpha_1, \ldots, \alpha_n \,\big]\,. \end{aligned}\right\} \tag{7.1}$$

(ii) For all $\tau \in \mu Tp$, $\alpha \in TVar$, and for all $\tau' \in \mu Tp$ such that $\tau' \equiv_{ren} \tau$ the following four equivalences hold:

$$\alpha \downarrow \tau \iff \alpha \notin fv(\tau) \;\vee\; \mathcal{L}'(\tau) = \rightarrow\;, \tag{7.2}$$

$$\alpha \downarrow \tau \iff \alpha \notin fv(\tau) \;\vee\; (\exists\, \rho_1, \rho_2 \in \mu Tp)\,[\, \tau \twoheadrightarrow_{ren/out\text{-}unf} \rho_1 \to \rho_2 \,]\,, \tag{7.3}$$

$$\alpha \downarrow \tau \iff nl\mu b(\,\tau'[\mu\alpha.\,\tau/\alpha]\,) < nl\mu b(\mu\alpha.\,\tau)\,, \tag{7.4}$$

$$\alpha \downarrow \tau \iff nl\mu b(\,\tau'[\mu\alpha.\,\tau/\alpha]\,) = nl\mu b(\mu\alpha.\,\tau) - 1\,. \tag{7.5}$$

Proof. We will show the items (i) and (ii) of the lemma in the below items (1) and (2). In both cases the definition of the expression $\alpha \downarrow \tau$, i.e. the assertion that τ is contractive with respect to α, from Definition 5.1.1 will be exploited, which, for all $\alpha \in TVar$ and $\tau \in \mu Tp$, can be put as:

$$\begin{aligned} \alpha \downarrow \tau \iff_{\text{def}} & \;\alpha \notin \text{fv}(\tau) \;\vee \\ & \vee\;\; (\exists\, n \in \omega)\,(\exists\, \alpha_1, \ldots, \alpha_n \in TVar) \\ & \qquad (\exists\, \rho_1, \rho_2 \in \mu Tp)\,\big[\, \tau \equiv \mu\alpha_1 \ldots \alpha_n.\,(\rho_1 \to \rho_2) \,\big]\,. \end{aligned} \tag{7.6}$$

(1) Let $\alpha \in TVar$ and $\tau \in \mu Tp$ be arbitrary, but in the following fixed. The statement (7.1) is demonstrated by the following chain of equivalences:

$$\begin{aligned} \alpha \not\downarrow \tau \Leftrightarrow & \neg\,\big[(\alpha \notin \text{fv}(\tau)) \;\vee \\ & \qquad \vee\;\; (\exists\, \rho_1, \rho_2)(\exists\, n \in \omega)(\exists\, \alpha_1 \ldots \alpha_n)\,[\,\tau \equiv \mu\alpha_1 \ldots \alpha_n.\,(\rho_1 {\to} \rho_2)\,]\big] \\ \Leftrightarrow & \;\; \alpha \in \text{fv}(\tau) \;\; \& \\ & \qquad \& \;\; (\forall\, \rho_1, \rho_2)(\forall\, n \in \omega)(\forall\, \alpha_1 \ldots \alpha_n)\,[\,\tau \not\equiv \mu\alpha_1 \ldots \alpha_n.\,(\rho_1 {\to} \rho_2)\,] \\ \Leftrightarrow & \; (\exists\, n \in \omega)(\exists\, \alpha_1, \ldots, \alpha_n \not\equiv \alpha)\,[\,\tau \equiv \mu\alpha_1 \ldots \alpha_n.\,\alpha\,] \end{aligned}$$

where clearly $\alpha_1 \dots \alpha_n$ range over type variables and ρ_1 and ρ_2 over recursive types in μTp. The first equivalence follows from the definition of $\alpha \downarrow \tau$, the second by an easy transformation from predicate logic. In the last equivalence the implication "$\Leftarrow$" is obvious; the implication "$\Rightarrow$" can be seen by an easy case-distinction argument, or more formally, by induction on $|\tau|$.

(2) Let $\alpha \in TVar$ and $\tau, \tau' \in \mu Tp$ be arbitrary such that $\tau' \equiv_{\text{ren}} \tau$ holds.

The equivalences (7.4) and (7.5) are immediate consequences of item (i) of the lemma and of Lemma 3.5.7.

Both of the equivalences (7.4) and (7.5) are obvious consequences of the definition of $\alpha \downarrow \tau$ and of the fact that, for all $\tau \in \mu Tp$, the following chain of equivalences

$$\left.\begin{aligned} \mathcal{L}'(\tau) = \to \quad &\underset{\text{(I)}}{\Longrightarrow} \quad (\exists\, \rho_1, \rho_2 \in \mu Tp)\,[\, \tau \twoheadrightarrow_{\text{ren/out-unf}} \rho_1 \to \rho_2 \,] \\ &\underset{\text{(II)}}{\Longrightarrow} \quad (\exists\, n \in \omega)\,(\exists\, \alpha_1, \dots, \alpha_n \in TVar) \\ &\qquad\qquad (\exists\, \rho_1, \rho_2 \in \mu Tp)\,[\, \tau \equiv \mu\alpha_1 \dots \alpha_n . (\rho_1 \to \rho_2) \,] \\ &\underset{\text{(III)}}{\Longrightarrow} \quad \mathcal{L}'(\tau) = \to \end{aligned}\right\} \tag{7.7}$$

holds. Therefore it suffices to justify (7.7). We will only indicate here how this can be done: The implication labeled by (I) can be shown by induction on the number $\text{nl}\mu\text{b}(\tau)$ of leading μ-bindings in τ using the definition $\mathcal{L}'(\tau) =_{\text{def}} \mathbf{Tree}(\tau)(\epsilon)$ of the leading-symbol function $\mathcal{L}'(\tau)$ and the definition of the tree unfolding of a recursive type. In a similar way, the implication labeled by (III) can be shown by an easy induction on n in which again the definitions of the leading symbol and of the tree unfolding of a recursive type are used. The implication labeled by (II) follows by induction on the length of a $\twoheadrightarrow_{\text{ren/out-unf}}$-reduction sequence from τ to $\rho_1 \to \rho_2$, where the induction step relies on the following auxiliary assertion

$$\begin{aligned} \tau \to_{\text{out-unf}} \mu\alpha_1 \dots \alpha_n . (\rho_1 \to \rho_2) \implies &(\exists\, \alpha \in TVar)\,(\exists\, \tilde{\rho}_1, \tilde{\rho}_2 \in \mu Tp) \\ &\big[\, \tau \equiv \mu\alpha\alpha_1 \dots \alpha_n . (\tilde{\rho}_1 \to \tilde{\rho}_2) \,\big] \end{aligned}$$

(for all $\tau, \rho_1, \rho_2 \in \mu Tp$ and $\alpha_1, \dots, \alpha_n \in TVar$) which can be verified by a quite straightforward analysis of all possible cases for τ_0, where $\tau \equiv \mu\alpha . \tau_0$ for some type variable α.

□

We will also need the following lemma.

Lemma 7.1.2. *(i) For all $\tau, \tau' \in \mu Tp$ and $\alpha \in TVar$:*

$$\alpha \downarrow \tau \;\;\&\;\; \tau' \equiv_{ren} \tau \implies \alpha \downarrow \tau' \,. \tag{7.8}$$

(ii) For all $\tau, \sigma \in \mu Tp$ and $\alpha, \beta \in TVar$:

$$\alpha \downarrow \tau \quad \& \quad \big(\alpha \notin fv(\tau) \;\Rightarrow\; \beta \equiv \alpha \;\vee\; \alpha \notin fv(\sigma)\big) \quad\Longrightarrow\quad \alpha \downarrow \tau[\sigma/\beta]\,. \qquad (7.9)$$

Proof. (1) For a proof of item (i) it suffices to show

$$(\forall\, \tau_1, \tau_2 \in \mu Tp)(\forall\, \beta \in TVar)\,[\,\beta \downarrow \tau_1 \quad \& \quad \tau_1 \to_{\text{ren}} \tau_2 \quad\Longrightarrow\quad \beta \downarrow \tau_2\,]\,. \quad (7.10)$$

This is because from this (7.8) can be shown, for all $\alpha \in TVar$ and $\tau, \tau' \in \mu Tp$ by induction on the length of a $\to_{\text{ren}}$-reduction sequence between τ and τ'.

To show (7.10), let $\tau_1, \tau_2 \in \mu Tp$ and a variable β be given such that $\beta \downarrow \tau_1$ and $\tau_1 \to_{\text{ren}} \tau_2$ hold.

Case 1: $\beta \notin \text{fv}(\tau_1)$: Then also $\beta \notin \text{fv}(\tau_2)$ and thus $\beta \downarrow \tau_2$ follows.

Case 2: $\tau_1 \equiv \mu\beta_1 \ldots \beta_n.\,(\rho_1 \to \rho_2)$ for some $n \in \omega$, $\beta_1, \ldots, \beta_n \in TVar$ and $\rho_1, \rho_2 \in \mu Tp$. Since $\tau_1 \to_{\text{ren}} \tau_2$, either

$$\tau_2 \equiv \mu\beta_1 \ldots \beta_{i-1}\tilde{\beta}\beta_{i+1} \ldots \beta_n.\,((\rho_1 \to \rho_2)[\tilde{\beta}/\beta_i])$$

must hold, for some $i \in \{1, \ldots, n\}$ and some variable $\tilde{\beta} \in TVar$, or

$$\tau_2 \equiv \mu\beta_1 \ldots \beta_n.\,(\rho_1' \to \rho_2')\,,$$

for some $\rho_1', \rho_2' \in \mu Tp$ with $\rho_i \to_{\text{ren}} \rho_i'$ and $\rho_{3-i}' \equiv \rho_{3-i}'$ for $i \in \{1, 2\}$. Then by the definition of $(\cdot \downarrow \cdot)$ in both situations $\beta \downarrow \tau_2$ follows.

In both cases the conclusion of (7.8) for the considered β, τ_1, τ_2 holds. Hence (7.8) has been shown, which concludes the proof of item (i).

(2) For proving (ii), let arbitrary $\alpha, \beta \in TVar$ and $\tau, \sigma \in \mu Tp$ be given such that $\alpha \downarrow \tau$ and

$$\alpha \notin \text{fv}(\tau) \;\Rightarrow\; \beta \equiv \alpha \;\vee\; \alpha \notin \text{fv}(\sigma)$$

holds, and such that σ is substitutible for α in τ. Due to the definition of $\alpha \downarrow \tau$ in Definition 5.1.1, it suffices to consider the following two cases separately.

Case 1: $\alpha \notin \text{fv}(\tau)$:

By assumption either $\beta \equiv \alpha$ holds, in which situation $\alpha \downarrow \tau[\sigma/\beta]$ follows by the assumption $\alpha \downarrow \tau$ and by $\tau[\sigma/\beta] \Leftrightarrow \tau$, or $\alpha \notin \text{fv}(\sigma)$ holds, which implies $\alpha \notin \text{fv}(\tau[\sigma/\beta])$ and hence also $\alpha \downarrow \tau[\sigma/\beta]$. Therefore we find here that $\alpha \downarrow \tau[\sigma/\beta]$ holds.

Case 2: $\tau \equiv \mu\alpha_1 \ldots \alpha_n.\,(\rho_1 \to \rho_2)$ for some $n \in \omega$, $\alpha_1, \ldots, \alpha_n \in TVar$ and $\rho_1, \rho_2 \in \mu Tp$:

If $\beta \notin \text{fv}(\tau)$, it holds that $\tau[\sigma/\beta] \Leftrightarrow \tau$, or, if $\beta \in \text{fv}(\tau)$, it holds that $\tau[\sigma/\beta] \Leftrightarrow \mu\alpha_1 \ldots \alpha_n.\,(\rho_1[\sigma/\beta] \to \rho_2[\sigma/\beta])$. In both situations $\alpha \downarrow$ $\downarrow \tau[\sigma/\beta]$ is implied (in the first by the assumption $\alpha \downarrow \tau$, in the second by the definition of $(\cdot \downarrow \cdot)$ in Definition 5.1.1).

In both cases we have concluded that $\alpha \downarrow \tau[\sigma/\beta]$ holds.

□

In the following definition we introduce a designation for the result of substituting a recursive type σ for a type variable α *throughout* an $\mathbf{AC}^{=}$-derivation $\mathcal{D}$ without assumptions, given that a certain condition holds on σ, α and $\mathcal{D}$. And the subsequent lemma states that the result of such a substitution is again an $\mathbf{AC}^{=}$-derivation without assumptions.

Definition 7.1.3 (Substitution in an $\mathbf{AC}^{=}$-derivation). Let $\tau_1, \tau_2 \in \mu Tp$, and let $\mathcal{D}$ be a derivation in $\mathbf{AC}^{=}$ without assumptions and with conclusion $\tau_1 = \tau_2$. Let furthermore $\alpha \in TVar$ and $\sigma \in \mu Tp$ be such that (1) for each occurrence of an equation between recursive types $\chi_1 = \chi_2$ in $\mathcal{D}$, σ is substitutible for α in both χ_1 and χ_2, and that (2) α is different from all type variables that are introduced as bound variables in $\mathcal{D}$ by an application of μ-COMPAT.

Then we define $\mathcal{D}[\sigma/\alpha]$ to be the result of replacing each occurrence of an equation between recursive types $\chi_1 = \chi_2$ in $\mathcal{D}$ by an occurrence of the equation between recursive types $\chi_1[\sigma/\alpha] = \chi_2[\sigma/\alpha]$. ⊠

Lemma 7.1.4 (Substitution in an $\mathbf{AC}^{=}$-derivation). *Let $\tau_1, \tau_2 \in \mu Tp$, and let $\mathcal{D}$ be a derivation in* $\mathbf{AC}^{=}$ *without assumptions and with conclusion $\tau_1 = \tau_2$. Let furthermore $\alpha \in TVar$ and $\sigma \in \mu Tp$ be such that the conditions (1) and (2) in Definition 7.1.3 are fulfilled.*

Then the formal object $\mathcal{D}[\sigma/\alpha]$ defined in Definition 7.1.3 is a derivation in $\mathbf{AC}^{=}$ *without assumptions and with conclusion $\tau_1[\sigma/\alpha] = \tau_2[\sigma/\alpha]$. The derivation $\mathcal{D}[\sigma/\alpha]$ has the same depth and size as $\mathcal{D}$.*

Proof. This can be shown by induction on $|\mathcal{D}|$, the depth of the derivation $\mathcal{D}$. The arguments for the necessary details, which have to be checked for this, are largely analogous to those in the proof below of Lemma 7.1.9, (1), but are much easier here.

Only one case shall be treated as an example here, a case for the induction step, in which the derivation ends with an application of the rule μ-COMPAT. For this, we let $\mathcal{D}$ be an $\mathbf{AC}^{=}$-derivation that does not contain assumptions and that is of the form

$$\frac{\begin{array}{c}\mathcal{D}_1\\ \tau_1 = \tau_2\end{array}}{\mu\beta.\,\tau_1 = \mu\beta.\,\tau_2}\ \mu\text{-COMPAT}$$

Furthermore, we let $\sigma \in \mu Tp$ and $\alpha \in TVar$ be such that the assumptions (1) and (2) in Definition 7.1.3 hold with respect to σ, α and $\mathcal{D}$.

By assumption (2) on α and $\mathcal{D}$ (which prevents α from being introduced in $\mathcal{D}$ by an application of μ-COMPAT) it follows that $\beta \not\equiv \alpha$. This implies that the two statements $\alpha \in \mathrm{fv}(\tau_1) \cup \mathrm{fv}(\tau_2)$ and $\alpha \in \mathrm{fv}(\mu\beta.\,\tau_1) \cup \mathrm{fv}(\mu\beta.\,\tau_2)$ are equivalent here. Therefore it suffices to prove the induction step only for the two separate cases $\alpha \in \mathrm{fv}(\mu\beta.\,\tau_1) \cup \mathrm{fv}(\mu\beta.\,\tau_2)$ and $\alpha \notin \mathrm{fv}(\tau_1) \cup \mathrm{fv}(\tau_2)$.

If $\alpha \in \mathrm{fv}(\mu\beta.\,\tau_1) \cup \mathrm{fv}(\mu\beta.\,\tau_2)$, then because of $\alpha \not\equiv \beta$ and σ is substitutible for α in $\mu\beta.\,\tau_i$ (for $i = 1, 2$) it holds that $(\mu\beta.\,\tau_i)[\sigma/\alpha] \Leftrightarrow \mu\beta.\,\tau_i[\sigma/\alpha]$ is the case for

$i \in \{1,2\}$. By the induction hypothesis, $\mathcal{D}_1[\sigma/\alpha]$ is an $\mathbf{AC}^{=}$-derivation without assumptions, with conclusion $\tau_1[\sigma/\alpha] = \tau_2[\sigma/\alpha]$, and with the same depth and the same size as $\mathcal{D}_1$. Then

$$\frac{\begin{array}{c}\mathcal{D}_1[\sigma/\alpha]\\ \tau_1[\sigma/\alpha] = \tau_2[\sigma/\alpha]\end{array}}{\underbrace{\mu\beta.\,\tau_1[\sigma/\alpha]}_{\Leftrightarrow\, (\mu\beta.\,\tau_1)[\sigma/\alpha]} \;=\; \underbrace{\mu\beta.\,\tau_2[\sigma/\alpha]}_{\Leftrightarrow\, (\mu\beta.\,\tau_2)[\sigma/\alpha]}}\;\mu\text{-COMPAT}$$

is an $\mathbf{AC}^{=}$-derivation without assumptions that conforms to the definition of $\mathcal{D}[\sigma/\alpha]$ and that has the same depth and the same size as $\mathcal{D}$.

If $\alpha \notin \mathrm{fv}(\tau_1) \cup \mathrm{fv}(\tau_2)$, then by the induction hypothesis $\mathcal{D}_1[\sigma/\alpha]$ is an $\mathbf{AC}^{=}$-derivation with conclusion $\tau_1 = \tau_2$, and with the same depth and the same size as $\mathcal{D}_1$. We furthermore have $\alpha \notin \mathrm{fv}(\mu\beta.\,\tau_1) \cup \mathrm{fv}(\mu\beta.\,\tau_2)$ in this case, which implies $\mu\beta.\,\tau_i \Leftrightarrow$ $\Leftrightarrow (\mu\beta.\,\tau_i)[\sigma/\alpha]$ for each $i \in \{1,2\}$. Hence the $\mathbf{AC}^{=}$-derivation

$$\frac{\begin{array}{c}\mathcal{D}_1[\sigma/\alpha]\\ \tau_1 = \tau_2\end{array}}{\mu\beta.\,\tau_1 = \mu\beta.\,\tau_2}\;\mu\text{-COMPAT}$$

without assumptions coincides with the definition of $\mathcal{D}[\sigma/\alpha]$; furthermore, it has the same depth and the same size as $\mathcal{D}$. □

Remark 7.1.5. The derivation $\mathcal{D}[\sigma/\alpha]$ has the same structure[1] and hence also the same depth and size as $\mathcal{D}$.

A generalization of Definition 7.1.3 and of the assertion of Lemma 7.1.4 is introduced and stated in the following Definition and Lemma, which we formulate as an aside; we will not use it subsequently.

Definition and Lemma 7.1.6. Let $\mathcal{D}$ be a derivation in $\mathbf{AC}^{=}$ without assumptions and with the conclusion $\tau_1 = \tau_2$, where $\tau_1, \tau_2 \in \mu Tp$, and let σ be a recursive type and α a type variable. Furthermore, suppose that σ is substitutible for α in all recursive types χ_1 and χ_2 for which there is an occurrence of an equality between recursive types $\chi_1 = \chi_2$ in $\mathcal{D}$ with the following property:

> The thread in $\mathcal{D}$ from the considered occurrence of $\chi_1 = \chi_2$ downwards to the conclusion of $\mathcal{D}$ does not cross any occurrence of such an application of the rule μ-COMPAT, in which α is introduced as a bound variable. } (7.11)

[1] That $\mathcal{D}$ and $\mathcal{D}[\sigma/\alpha]$ possess the same *structure* is intended to refer to a statement that could be made precise as follows. We consider the prooftrees $\mathcal{D}$ and $\mathcal{D}[\sigma/\alpha]$ as labeled trees, where with respect to the underlying unlabeled trees equations between recursive types are assigned to the respective nodes and where names of rules are assigned to their respective edges (according to the corresponding rule applications that relate (the "nodes" of) corresponding premises and conclusions in the representations of $\mathcal{D}$ and $\mathcal{D}[\sigma/\alpha]$ as graphical prooftrees—as these are always used here and elsewhere in this paper). Then the underlying unlabeled trees of $\mathcal{D}$ and $\mathcal{D}[\sigma/\alpha]$ are identical and furthermore the labels for the *edges* (which are names of rules of $\mathbf{AC}^{=}$) coincide in $\mathcal{D}$ and $\mathcal{D}[\sigma/\alpha]$.

Then we denote by $\mathcal{D}([\sigma/\alpha])$ the result of replacing in the prooftree $\mathcal{D}$ each occurrence of an equation between recursive types $\chi_1 = \chi_2$ with the property (7.11) by an occurrence of $\chi_1[\sigma/\alpha] = \chi_2[\sigma/\alpha]$. It holds that $\mathcal{D}([\sigma/\alpha])$ is an derivation in $\mathbf{AC}^{=}$ without assumptions and with the conclusion $\tau_1[\sigma/\alpha] = \tau_2[\sigma/\alpha]$ that has the same depth and the same size as $\mathcal{D}$. ⊠

The *proof* of the 'lemma-part' of Definition and Lemma 7.1.6 is an easy refinement of the proof of Lemma 7.1.4.

Now we define three kinds of substitution rules on the set of equations between recursive types. In the subsequent remark we discuss a slight conceptual difference, hinging on our use of Convention 3.3.6 in dealing with substitution expressions, between substitution rules introduced, on the one hand, in the items (i) and (iii), and on the other hand, in item (ii) of the definition below.

Definition 7.1.7 (Three kinds of substitution rules). We define the following three kinds of *substitution rules* for pure Hilbert-systems that have the set $\mu Tp\text{–}Eq$ as their set of formulas.

(i) For all $\sigma \in \mu Tp$ and $\alpha \in TVar$, the rule $\mathrm{SUBST}_{(.)[\sigma/\alpha]}$ has precisely those applications on the set $\mu Tp\text{–}Eq$ that are schematically defined by

$$\frac{\tau_1 = \tau_2}{\tau_1'[\sigma/\alpha] = \tau_2'[\sigma/\alpha]}\ \mathrm{SUBST}_{(.)'[\sigma/\alpha]} \quad (\text{if } \tau_1' \equiv_{\mathrm{ren}} \tau_1 \text{ and } \tau_2' \equiv_{\mathrm{ren}} \tau_2) \tag{7.12}$$

where $\tau_1, \tau_2, \tau_1', \tau_2' \in \mu Tp$.

(ii) For all $\tau \in \mu Tp$ and $\alpha \in TVar$, the rule $\mathrm{SUBST}_{\tau[./\alpha]}$ has precisely the applications that are schematically defined by

$$\frac{\sigma_1 = \sigma_2}{\tau[\sigma_1/\alpha] = \tau[\sigma_2/\alpha]}\ \mathrm{SUBST}_{\tau[./\alpha]} \tag{7.13}$$

where $\sigma_1, \sigma_2 \in \mu Tp$.

(iii) For all $\alpha \in TVar$, the rule $\mathrm{SUBST}_{(.)'[./\alpha]}$ has precisely the applications of the scheme

$$\frac{\tau_1 = \tau_2 \qquad \sigma_1 = \sigma_2}{\tau_1'[\sigma_1/\alpha] = \tau_2'[\sigma_2/\alpha]}\ \mathrm{SUBST}_{(.)'[./\alpha]} \quad (\text{if } \tau_1' \equiv_{\mathrm{ren}} \tau_1 \text{ and } \tau_2' \equiv_{\mathrm{ren}} \tau_2) \tag{7.14}$$

where $\tau_1, \tau_2, \tau_1', \tau_2', \sigma_1, \sigma_2 \in \mu Tp$.

⊠

Remark 7.1.8. We want to stress here the obvious consequence that our use of Convention 3.3.6 has for the above definition of three kinds of substitution rules: the defining schemes (7.12), (7.13), and (7.14) for these rules are subject to the implicit side-conditions on the substitution expressions in the conclusions of these schemes. For instance, item (ii) of Definition 7.1.7 has to be understood as follows: For all $\tau, \sigma_1, \sigma_2 \in \mu Tp$ and $\alpha \in TVar$, the applications of the substitution rules $\mathrm{SUBST}_{\tau[./\alpha]}$ are, together with their respective premises and conclusions, defined by those corresponding inference figures of the form (7.13) for which the substitution expressions $\tau[\sigma_1/\alpha]$ and $\tau[\sigma_2/\alpha]$ occurring in its conclusion are admissible, i.e. for which σ_1 and σ_2 are substitutible for α in τ.

In relation with this, we furthermore notice that there is a slight conceptual difference between substitution rules of the kinds (i) and (iii) of Definition 7.1.7 on the one hand and substitution rules of kind (ii) in Definition 7.1.7 on the other hand. In the case of a rule $\mathrm{SUBST}_{(.)'[\sigma/\alpha]}$, with some $\sigma \in \mu Tp$ and $\alpha \in TVar$, there exists, for every equation between recursive types $\tau_1 = \tau_2$, an application of $\mathrm{SUBST}_{(.)'[\sigma/\alpha]}$ with $\tau_1 = \tau_2$ as premise (in fact there typically exist infinitely many such applications); and similarly, in the case of a rule $\mathrm{SUBST}_{(.)'[./\alpha]}$ there exists, for every pair $\tau_1 = \tau_2$ and $\sigma_1 = \sigma_2$ of equations between recursive types, an application of this rule with the premises $\tau_1 = \tau_2$ and $\sigma_1 = \sigma_2$ (again, there will typically be infinitely many such applications). However, in the case of a rule $\mathrm{SUBST}_{\tau[./\alpha]}$ not every equation between recursive types $\sigma_1 = \sigma_2$ is the premise of an application of this rule because σ_1 and σ_2 do not need to be substitutible for α in τ. We could alternatively have introduced generalizations $\mathrm{SUBST}_{(\tau)'[./\alpha]}$ of the substitution rules $\mathrm{SUBST}_{\tau[./\alpha]}$, where, for all $\tau \in \mu Tp$ and $\alpha \in TVar$, the rule $\mathrm{SUBST}_{(\tau)'[./\alpha]}$ is defined by the scheme of inferences

$$\frac{\sigma_1 = \sigma_2}{\tau'[\sigma_1/\alpha] = \tau''[\sigma_2/\alpha]}\ \mathrm{SUBST}_{(\tau)'[./\alpha]} \quad (\text{if } \tau' \equiv_{\mathrm{ren}} \tau \text{ and } \tau'' \equiv_{\mathrm{ren}} \tau) \tag{7.15}$$

in which σ_1 and σ_2 vary over recursive types in μTp. All of these variant rules have again the property that they are applicable on every equation between recursive types.

We have chosen not to introduce the variant rules $\mathrm{SUBST}_{(\tau)'[./\alpha]}$ instead of the rules $\mathrm{SUBST}_{\tau[./\alpha]}$ in Definition 7.1.7 for the following three reasons: (a) for all $\tau \in \mu Tp$ and every equation $\sigma_1 = \sigma_2 \in \mu Tp\text{–}Eq$ there exists a variant $\tau' \in \mu Tp$ such that the rule $\mathrm{SUBST}_{\tau'[./\alpha]}$ *is applicable* on $\sigma_1 = \sigma_2$, (b) the variant rules $\mathrm{SUBST}_{(\tau)'[./\alpha]}$ can be viewed as special cases of substitution rules of Definition 7.1.7, (iii) (this holds clearly also for rules of the kinds (i) and (ii) in this definition), and (c) Lemma 7.1.9 below can be shown slightly easier with respect to rules $\mathrm{SUBST}_{\tau[./\alpha]}$ (although it holds also with respect to the rules $\mathrm{SUBST}_{(\tau)'[./\alpha]}$, see the statement of Proposition 7.1.11 given later).

The following lemma will later be used as the essential tool for the purpose of recognizing that μ-COMPAT is an admissible rule of a variant system close to $\mathbf{AC}^{=}$–{μ-COMPAT}.[2] It asserts admissibility, and in one case even derivability, in $\mathbf{AC}^{=}$ of the substitution rules defined in Definition 7.1.7. Admissibility in $\mathbf{AC}^{=}$ of substitution rules from item (ii) in Definition 7.1.7 is used tacitly and without

[2] More precisely, μ-COMPAT will be seen to be an admissible rule of $((\mathbf{AC}^{=} \setminus \{\mu\text{-COMPAT} + (\mu-\bot)\}) + (\mu-\bot)')$, where as axioms of the scheme $(\mu-\bot)'$ similar, but slightly more general formulas than the axioms $(\mu-\bot)$ from $\mathbf{AC}^{=}$ will be allowed.

proof in [AmCa93] (cf. Remark 7.1.10 below); this statement, however, turns out to be the easiest item of the following lemma to prove.

Lemma 7.1.9 (Admissibility of substitution rules in $\mathbf{AC}^{=}$). *The following assertions hold for the three kinds of substitution rules defined in Definition 7.1.7:*

(i) For all $\sigma \in \mu Tp$ and $\alpha \in TVar$, the rule $\mathrm{SUBST}_{(.)'[\sigma/\alpha]}$ *is admissible in* $\mathbf{AC}^{=}$.

(ii) For all $\tau \in \mu Tp$ and $\alpha \in TVar$, the rule $\mathrm{SUBST}_{\tau[./\alpha]}$ *is derivable, and hence also admissible, in* $\mathbf{AC}^{=}$.

(iii) For all $\alpha \in TVar$, the rule $\mathrm{SUBST}_{(.)'[./\alpha]}$ *is admissible in* $\mathbf{AC}^{=}$.

Moreover, every derivation $\mathcal{D}$ without assumptions in the extension of $\mathbf{AC}^{=}$ *with the substitution rules from Definition 7.1.7 can effectively be transformed into an* $\mathbf{AC}^{=}$*-derivation $\mathcal{D}'$ without assumptions and with the same conclusion by a finite sequence of successively applied, local manipulation-steps.*

Proof. We will argue in item (2) below why the substitution rules of the kind (ii) in Definition 7.1.7 are derivable rules of $\mathbf{AC}^{=}$. Apart from this, it will suffice to show the assertion of the lemma that applications of substitution rules from Definition 7.1.7, (i), (ii), and (ii), can effectively be eliminated because this obviously implies that these rules are admissible rules in $\mathbf{AC}^{=}$.

The possibility to effectively eliminate applications of substitution rules from a derivation $\tilde{\mathcal{D}}$, without assumptions, in the extension of $\mathbf{AC}^{=}$ by adding the substitution rules defined in Definition 7.1.7 follows by induction on the number of applications of such rules in $\tilde{\mathcal{D}}$, once it will be proven that:

Every derivation $\mathcal{D}$ that contains no assumptions, and that is of the form $\dfrac{\mathcal{D}_1 \qquad (\mathcal{D}_1)}{\chi_1 = \chi_2}\,R$, where $\mathcal{D}_1$ and (if it indeed also appears in $\mathcal{D}$) $\mathcal{D}_1$ are $\mathbf{AC}^{=}$-derivations without assumptions and where R is an application of a substitution rule of the kind (i), (ii) or (iii) in Definition 7.1.7 can effectively be transformed into an $\mathbf{AC}^{=}$-derivation $\mathcal{D}'$ without assumptions and with conclusion $\chi_1 = \chi_2$. (7.16)

Given a derivation $\tilde{\mathcal{D}}$ without assumptions in the extension of $\mathbf{AC}^{=}$ with the substitution rules from Definition 7.1.7, we can then proceed, analogously to traditional proofs for cut-elimination in sequent calculi, as follows. By using (7.16), every subderivation $\tilde{\mathcal{D}}_1$ of $\tilde{\mathcal{D}}$ that ends in a *topmost* occurrence of a substitution rule can be replaced by an $\mathbf{AC}^{=}$-derivation $\tilde{\mathcal{D}}_1'$ without assumptions and with the same respective conclusion. Any such replacement reduces the number of substitution rules in the derivation by one. It is clear that by continuing to eliminate topmost occurrences of substitution rules in $\tilde{\mathcal{D}}^{(1)}$, the result of a first replacement of such a respective subderivation, we are lead, after eliminating all substitution rules that occur in $\tilde{\mathcal{D}}$, to a derivation $\tilde{\mathcal{D}}'$ in $\mathbf{AC}^{=}$ without assumptions and with the same conclusion as $\tilde{\mathcal{D}}$.

Hence it suffices to prove (7.16). For this we will prove separately, in the items (1), (2) and (3) below, that parts of (7.16) which refer to the three kinds of substitution rules from Definition 7.1.7, respectively.

(1) Here we will prove that part of (7.16) which refers to substitution rules R of the form $\mathrm{SUBST}_{(.)'[\sigma/\alpha]}$, where $\sigma \in \mu Tp$ and $\alpha \in TVar$ are arbitrary.

For this, we let $\sigma \in \mu Tp$ and $\alpha \in TVar$ be arbitrary, but in the following fixed. We have to show that every derivation $\mathcal{D}$ of the form

$$\frac{\begin{array}{c}\mathcal{D}_1\\ \tau_1 = \tau_2\end{array}}{\tau_1'[\sigma/\alpha] = \tau_2'[\sigma/\alpha]}\ \mathrm{SUBST}_{(.)'[\sigma/\alpha]} \tag{7.17}$$

where $\tau_1, \tau_2, \tau_1', \tau_2' \in \mu Tp$, $\tau_1' \equiv_{\mathrm{ren}} \tau_1$ and $\tau_2' \equiv_{\mathrm{ren}} \tau_2$, and where $\mathcal{D}_1$ is a derivation in $\mathbf{AC}^{=}$ without assumptions, can effectively be transformed into a derivation $\mathcal{D}'$ in $\mathbf{AC}^{=}$ without assumptions and with the same conclusion.

We will proceed by induction on $|\mathcal{D}_1|$. The procedure for eliminating applications of $\mathrm{SUBST}_{(.)'[\sigma/\alpha]}$ implicit in the ensuing proof resembles many traditional cut-elimination procedures for derivations containing applications of the cut-rule in sequent calculi for classical or intuitionistic logic: topmost applications of substitution rules $\mathrm{SUBST}_{(.)'[\sigma/\alpha]}$ in a derivation $\bar{\mathcal{D}}$ without assumptions can be eliminated directly whenever they immediately follow axioms in $\bar{\mathcal{D}}$; if topmost $\mathrm{SUBST}_{(.)'[\sigma/\alpha]}$-applications succeed rule applications, then they can be stepwisely permuted upwards over all preceding rule applications in $\bar{\mathcal{D}}$ until they immediately follow axioms (where they can then again be always replaced either by other axioms or by short $\mathbf{AC}^{=}$-derivations).

To consider the base case of the induction, we assume an arbitrary derivation of the form (7.17) to be given, where $\mathcal{D}_1$ is an $\mathbf{AC}^{=}$-derivation without assumptions and with $|\mathcal{D}_1| = 0$. This means that $\mathcal{D}_1$ consists just of an $\mathbf{AC}^{=}$-axiom. We distinguish the cases of the different kinds of axioms of $\mathbf{AC}^{=}$.

If $\mathcal{D}$ is of the form $\dfrac{\dfrac{(\mathrm{REFL})}{\tau = \tau}}{\tau'[\sigma/\alpha] = \tau''[\sigma/\alpha]}\ {\scriptstyle \mathrm{SUBST}_{(.)'[\sigma/\alpha]}}$, and hence if $\mathcal{D}_1$ is an axiom (REFL) of $\mathbf{AC}^{=}$, then by Lemma 3.4.2, the assertion associated with (3.18), the conclusion of $\mathcal{D}$ is an axiom (REN) of $\mathbf{AC}^{=}$, which can be taken as the desired transformed derivation $\mathcal{D}'$ in $\mathbf{AC}^{=}$ without assumptions.

Similarly in the case, where $\mathcal{D}_1$ is an axiom (REN), the conclusion of $\mathcal{D}$ is again an axiom (REN), which can be taken to be the transformed derivation $\mathcal{D}'$. If $\mathcal{D}_1$ is an axiom $(\mu - \bot)$, then also the conclusion of $\mathcal{D}$ is such an axiom, which can again be taken as $\mathcal{D}'$.

If $\mathcal{D}_1$ is an axiom (FOLD/UNFOLD), then $\mathcal{D}$ is of the form

$$\dfrac{\dfrac{\text{(FOLD/UNFOLD)}}{\mu\beta.\,\tau_0 = \tau_0[\mu\beta.\,\tau_0/\beta]}}{(\mu\beta.\,\tau_0)'\,[\sigma/\alpha] = (\tau_0[\mu\beta.\,\tau_0/\beta])'\,[\sigma/\alpha]}\ \mathrm{SUBST}_{(.)'[\sigma/\alpha]}$$

where the primed subexpressions designate two respective variants of the recursive types denoted by the expressions in the brackets that allow the substitution of σ for α. If $\alpha \notin \mathrm{fv}(\mu\beta.\,\tau_0)$ is the case, then the application of $\mathrm{SUBST}_{(.)'[\sigma/\alpha]}$ corresponds just to taking variants on both sides of the equation in the premise; therefore, this application can be replaced by (at most) two successive applications of REN_r and REN_l, which can be eliminated easily in their turn. Hence we can assume now that $\alpha \in \mathrm{fv}(\mu\beta.\,\tau_0)$ holds. As a consequence, we find that $\alpha \in \mathrm{fv}(\tau_0)$ and $\alpha \not\equiv \beta$ hold.

Now let τ_0' be such a variant of τ_0 that, on the one hand, σ is substitutible for both α and β in τ_0' and that, on the other hand, $\mu\beta.\,\tau_0$ is substitutible for β in τ_0'. Let further $\tilde{\beta}$ be a variable such that ($\tilde{\beta} \equiv \beta$ or $\tilde{\beta} \notin \mathrm{fv}(\tau_0')$), $\tilde{\beta} \notin \mathrm{fv}(\sigma)$ and $\tilde{\beta}$ is substitutible for β in τ_0'.

Then $\tilde{\beta} \not\equiv \alpha$ holds (which follows by the choice of $\tilde{\beta}$ from $\alpha \in \mathrm{fv}(\tau_0) = \mathrm{fv}(\tau_0')$ and $\beta \not\equiv \alpha$), and that $\mu\tilde{\beta}.\,\tau_0'[\tilde{\beta}/\beta]$ is a variant of $\mu\beta.\,\tau_0$. Furthermore it follows that each of the four substitution expressions $\tau_0'[\tilde{\beta}/\beta]\,[\sigma/\alpha]$, $\mu\tilde{\beta}.\,\tau_0'[\tilde{\beta}/\beta]\,[\sigma/\alpha]$, $\tau_0'[\tilde{\beta}/\beta]\,[\mu\tilde{\beta}.\,\tau_0'[\tilde{\beta}/\beta]/\tilde{\beta}]$, and $(\tau_0'[\tilde{\beta}/\beta]\,[\sigma/\alpha])\,[\mu\tilde{\beta}.\,\tau_0'[\tilde{\beta}/\beta]\,[\sigma/\alpha]/\tilde{\beta}]$ are admissible. Given this knowledge, the derivation $\tilde{\mathcal{D}}$ of the form

$$\begin{array}{c}
\text{(FOLD/UNFOLD)} \\ \hline
\underbrace{\mu\tilde{\beta}.\,\tau_0'[\tilde{\beta}/\beta]\,[\sigma/\alpha]}_{\underset{\text{(I)}}{\Leftrightarrow} \underbrace{(\mu\tilde{\beta}.\,\tau_0'[\tilde{\beta}/\beta])}_{\equiv_{\mathrm{ren}}\ \mu\beta.\,\tau_0}\,[\sigma/\alpha]} = \underbrace{(\tau_0'[\beta/\tilde{\beta}]\,[\sigma/\alpha])\,[\underbrace{\mu\tilde{\beta}.\,\tau_0'[\tilde{\beta}/\beta]\,[\sigma/\alpha]}_{\underset{\text{(I)}}{\Leftrightarrow} (\mu\tilde{\beta}.\,\tau_0'[\tilde{\beta}/\beta])\,[\sigma/\alpha]}\ /\tilde{\beta}]}_{\underset{\text{(II)}}{\Leftrightarrow} \underbrace{(\tau_0'[\tilde{\beta}/\beta]\,[\mu\tilde{\beta}.\,\tau_0'[\tilde{\beta}/\beta]/\tilde{\beta}])}_{\underset{\text{(III)}}{\Leftrightarrow} \tau_0'[\underbrace{\mu\tilde{\beta}.\,\tau_0'[\beta/\tilde{\beta}]}_{\equiv_{\mathrm{ren}}\ \mu\beta.\,\tau_0}\,/\beta]}\,[\sigma/\alpha]} \\ \hline\hline
(\mu\beta.\,\tau_0)'\,[\sigma/\alpha] = (\tau_0[\mu\beta.\,\tau_0/\beta])'\,[\sigma/\alpha]
\end{array}\ \mathrm{REN}_l,\mathrm{REN}_r$$

can be built. Thereby the equality equivalence (III) is obvious, the equivalences (I) follow from Lemma 3.3.10, (iii), due to $\alpha \not\equiv \tilde{\beta}$ and $\tilde{\beta} \notin \mathrm{fv}(\sigma)$ and (II) is due to Lemma 3.3.11, (iii), since we know the admissibility of both substitution expressions in question. The justifications $\tau_0'\,[(\mu\tilde{\beta}.\,\tau_0'[\beta/\tilde{\beta}])/\beta] \equiv_{\mathrm{ren}}$ $\equiv_{\mathrm{ren}} (\tau_0[\mu\beta.\,\tau_0/\beta])'\,[\sigma/\alpha]$ and $(\mu\beta.\,\tau_0'[\tilde{\beta}/\beta])\,[\sigma/\alpha] \equiv_{\mathrm{ren}} (\mu\beta.\,\tau_0)'\,[\sigma/\alpha]$ for the applications $\mathrm{REN}_{l/r}$ at the bottom of $\tilde{\mathcal{D}}$ follow from Lemma 3.4.2, (3.18). Finally the derivation $\tilde{\mathcal{D}}$ can easily be transformed into an $\mathbf{AC}^{=}$-derivation $\mathcal{D}'$ without assumptions by eliminating the $\mathrm{REN}_{l/r}$-applications.

For carrying out the induction step, we let $\mathcal{D}$ be an arbitrary derivation of the form (7.17), where $\mathcal{D}_1$ is an $\mathbf{AC}^{=}$-derivation without assumptions such that $|\mathcal{D}_1| > 0$ holds. We will use case-distinction according to which rule of $\mathbf{AC}^{=}$

is applied at the bottom of $\mathcal{D}_1$, and we will always proceed by permuting the application of $\text{SUBST}_{(.)'[\sigma/\alpha]}$ at the bottom of $\mathcal{D}$ upwards over the last rule application in $\mathcal{D}_1$. In all cases we will be able to use the induction hypothesis in order to build a transformed **AC$^=$**-derivation $\mathcal{D}'$ without assumptions and with the same conclusion as $\mathcal{D}$.

If the last rule application in the **AC$^=$**-derivation $\mathcal{D}_1$ is a SYMM-rule, then $\mathcal{D}$ is of the form

$$\dfrac{\dfrac{\begin{array}{c}\mathcal{D}_{11}\\ \tau_2 = \tau_1\end{array}}{\tau_1 = \tau_2}\,\text{SYMM}}{\tau_1'[\sigma/\alpha] = \tau_2'[\sigma/\alpha]}\,\text{SUBST}_{(.)'[\sigma/\alpha]}$$

where $\tau_1' \equiv_{\text{ren}} \tau_1$ and $\tau_2' \equiv_{\text{ren}} \tau_2$. This derivation can be transformed into a derivation $\tilde{\mathcal{D}}$ of the form

$$\dfrac{\dfrac{\begin{array}{c}\mathcal{D}_{11}\\ \tau_2 = \tau_1\end{array}}{\tau_2'[\sigma/\alpha] = \tau_1'[\sigma/\alpha]}\,\text{SUBST}_{(.)'[\sigma/\alpha]}}{\tau_1'[\sigma/\alpha] = \tau_2'[\sigma/\alpha]}\,\text{SYMM}$$

Since $|\mathcal{D}_{11}| < |\mathcal{D}_1|$ holds, the induction hypothesis entails that the subderivation $\tilde{\mathcal{D}}_1$ of $\tilde{\mathcal{D}}$ ending with the displayed $\text{SUBST}_{(.)'[\sigma/\alpha]}$-application can effectively be transformed into an **AC$^=$**-derivation $\mathcal{D}_1'$ with the same conclusion, which then can replace $\tilde{\mathcal{D}}_1$ in $\tilde{\mathcal{D}}$ to arrive at an **AC$^=$**-derivation $\mathcal{D}'$ without assumptions and with the same conclusion as $\mathcal{D}$.

The cases, in which the last rule application in $\mathcal{D}_1$ is an application of TRANS or of ARROW can be settled quite analogously.

If the last rule application in $\mathcal{D}_1$ is a μ-COMPAT-rule, then $\mathcal{D}$ is of the form

$$\dfrac{\dfrac{\begin{array}{c}\mathcal{D}_{11}\\ \tau_1 = \tau_2\end{array}}{\mu\beta.\tau_1 = \mu\beta.\tau_2}\,\mu\text{-COMPAT}}{(\mu\beta.\tau_1)'\,[\sigma/\alpha] = (\mu\beta.\tau_2)'\,[\sigma/\alpha]}\,\text{SUBST}_{(.)'[\sigma/\alpha]}$$

where $(\mu\beta.\tau_1)' \equiv_{\text{ren}} \mu\beta.\tau_1$ and $(\mu\beta.\tau_2)' \equiv_{\text{ren}} \mu\beta.\tau_2$. If $\alpha \notin \text{fv}(\mu\beta.\tau_1)$ and $\alpha \notin$ $\notin \text{fv}(\mu\beta.\tau_2)$, the $\text{SUBST}_{(.)'[\sigma/\alpha]}$-application at the bottom of $\mathcal{D}$ only amounts to taking variants on either side of the equation in its premise. Then this rule application can be replaced by one or by two applications of rules $\text{REN}_{l/r}$ that can be easily eliminated in their turn to arrive at a derivation $\mathcal{D}'$ in **AC$^=$** without assumptions and with the same conclusion as $\mathcal{D}$. Next we consider the case that α occurs free in at least one of the recursive types $\mu\beta.\,\tau_1$ and $\mu\beta.\,\tau_2$.

Suppose now, that either $\alpha \in \mathrm{fv}(\mu\beta.\,\tau_1)$ or $\alpha \in \mathrm{fv}(\mu\beta.\,\tau_2)$ holds. Then clearly $\alpha \not\equiv \beta$ and $\alpha \in \mathrm{fv}(\tau_1) \cup \mathrm{fv}(\tau_2)$. Let now τ_1' and τ_2' be respective variants of τ_1 and τ_2 such that σ is substitutible for α in τ_1' and τ_2'. Furthermore let $\tilde{\beta}$ be a variable neither occurring in (any recursive type of) $\mathcal{D}_{11}$ nor in τ_1' or τ_2', such that also $\tilde{\beta} \notin \mathrm{fv}(\sigma)$ holds. Then $\tilde{\beta} \not\equiv \alpha$ and $\tilde{\beta}$ is substitutible for β in τ_1' and τ_2' and σ is substitutible for α in $\tau_1'[\tilde{\beta}/\beta]$ and in $\tau_2'[\tilde{\beta}/\beta]$. Moreover for $i \in \{1,2\}$ it holds that $\mu\tilde{\beta}.\,\tau_i'[\tilde{\beta}/\beta]$ is a variant of $\mu\beta.\,\tau_i$. Since $\tilde{\beta}$ does not occur in $\mathcal{D}_{11}$ at all, it follows by Lemma 7.1.4 that $\mathcal{D}_{11}[\tilde{\beta}/\beta]$ is an $\mathbf{AC}^{=}$-derivation without assumptions, with conclusion $\tau_1'[\tilde{\beta}/\beta] = \tau_2'[\tilde{\beta}/\beta]$, and with $|\mathcal{D}_{11}[\tilde{\beta}/\beta]| = |\mathcal{D}_{11}|$. Hence $\mathcal{D}$ can be transformed into the derivation $\tilde{\mathcal{D}}$ of the form

$$\begin{array}{cl}
\mathcal{D}_{11}[\tilde{\beta}/\beta] & \\
\tau_1[\tilde{\beta}/\beta] = \tau_2[\tilde{\beta}/\beta] & \\ \hline
\overbrace{\tau_1'[\tilde{\beta}/\beta]}^{\underset{(\mathrm{A})}{\equiv_{\mathrm{ren}}}\ \tau_1[\tilde{\beta}/\beta]}\ [\sigma/\alpha] = \overbrace{\tau_2'[\tilde{\beta}/\beta]}^{\underset{(\mathrm{B})}{\equiv_{\mathrm{ren}}}\ \tau_2[\tilde{\beta}/\beta]}\ [\sigma/\alpha] & \mathrm{SUBST}_{(.)'[\sigma/\alpha]} \\ \hline
\underbrace{\mu\tilde{\beta}.\,\tau_1'[\tilde{\beta}/\beta]\,[\sigma/\alpha]}_{\underset{(\mathrm{I})}{\Longleftrightarrow}\ (\underbrace{\mu\tilde{\beta}.\,\tau_1'[\tilde{\beta}/\beta]}_{\equiv_{\mathrm{ren}}\ \mu\beta.\,\tau_1})\,[\sigma/\alpha]} = \underbrace{\mu\tilde{\beta}.\,\tau_2'[\tilde{\beta}/\beta]\,[\sigma/\alpha]}_{\underset{(\mathrm{II})}{\Longleftrightarrow}\ (\underbrace{\mu\tilde{\beta}.\,\tau_2'[\tilde{\beta}/\beta]}_{\equiv_{\mathrm{ren}}\ \mu\beta.\,\tau_2})\,[\sigma/\alpha]} & \mu\text{-COMPAT} \\ \hline\hline
(\mu\beta.\,\tau_1)'\,[\sigma/\alpha] = (\mu\beta.\,\tau_2)'\,[\sigma/\alpha] & \mathrm{REN}_l,\ \mathrm{REN}_r
\end{array}$$

that does not contain assumptions. The equality equivalences (I) and (II) are justified by Lemma 3.3.10, (iii), which can be applied in each case due to the facts $\alpha \not\equiv \tilde{\beta}$ and $\tilde{\beta} \notin \mathrm{fv}(\sigma)$; for (A) and (B) assertion associated with (3.18) in Lemma 3.4.2 has been used. Now the induction hypothesis can be applied to the subderivation of $\tilde{\mathcal{D}}$ that ends in the displayed application of $\mathrm{SUBST}_{(.)'[\sigma/\alpha]}$. Hence this subderivation of $\tilde{\mathcal{D}}$ can be replaced by an $\mathbf{AC}^{=}$-derivation without assumptions that moreover is the result of an effective transformation-process applied to $\mathcal{D}_{11}$. After eliminating the final applications of $\mathrm{REN}_{l/r}$ from the arising derivation (by replacing these by applications of TRANS with axioms (REN) in an obvious way) an $\mathbf{AC}^{=}$-derivation $\mathcal{D}'$ without assumptions and with the same conclusion as $\mathcal{D}$ has effectively been produced.

If the last rule application in $\mathcal{D}_1$ is UFP, then $\mathcal{D}$ is of the form

$$\dfrac{\dfrac{\begin{array}{c}\mathcal{D}_{11} \\ \tau_1 = \tau[\tau_1/\beta]\end{array} \qquad \begin{array}{c}\mathcal{D}_{12} \\ \tau_2 = \tau[\tau_2/\beta]\end{array}}{\tau_1 = \tau_2}\ \mathrm{UFP}}{\tau_1'[\sigma/\alpha] = \tau_2'[\sigma/\alpha]}\ \mathrm{SUBST}_{(.)'[\sigma/\alpha]}$$

We will subsequently restrict our attention to the case that the condition $\beta \in \mathrm{fv}(\tau)$ is fulfilled with respect to the application of UFP displayed in the above prooftree for $\mathcal{D}$. If to the contrary $\beta \notin \mathrm{fv}(\tau)$ holds, we can argue as follows: Then the application UFP at the bottom of $\mathcal{D}_1$ can be replaced by an application of SYMM at the bottom of $\mathcal{D}_{12}$ followed by an application of TRANS with $\mathcal{D}_{11}$ leading up to its left premise. Now the application of SUBST$_{(.)'[\sigma/\alpha]}$ at the bottom of $\mathcal{D}$ can be permuted upwards over these two rule applications in the way discussed earlier here and then the induction hypothesis can be applied for two respective subderivations.

Apart from $\beta \notin \mathrm{fv}(\tau)$, we will furthermore assume that $\beta \notin \mathrm{fv}(\sigma)$ and $\beta \not\equiv \alpha$ are true: If namely $\beta \in \mathrm{fv}(\sigma)$ or $\beta \equiv \alpha$ were the case, then we would be able to replace τ by $\tau[\tilde{\beta}/\beta]$ for some variable $\tilde{\beta}$ with the properties $\tilde{\beta} \notin \mathrm{fv}(\sigma)$, $\tilde{\beta} \not\equiv \alpha$, $\tilde{\beta} \notin \mathrm{fv}(\tau)$ and $\tilde{\beta}$ substitutible for β in τ and find that the application of UFP at the bottom of $\mathcal{D}_1$ is justified because it is also of the form

$$\dfrac{\tau_1 = \overbrace{(\tau[\tilde{\beta}/\beta])\,[\tau_1/\tilde{\beta}]}^{\Leftrightarrow\ \tau[\tau_1/\beta]} \qquad \tau_2 = \overbrace{(\tau[\tilde{\beta}/\beta])\,[\tau_2/\tilde{\beta}]}^{\Leftrightarrow\ \tau[\tau_2/\beta]}}{\tau_1 = \tau_2}\ \mathrm{UFP}$$

Now let (a) τ_1' and τ_2' be variants of τ_1 and τ_2, respectively, such that σ is substitutible for α in τ_1' and τ_2', and (b) τ' be a variant of τ such that τ_1' and τ_2' are substitutible for β in τ' and σ is substitutible for α and β in τ'. Due to this and because of $\beta \notin \mathrm{fv}(\sigma)$ we find that $\tau_1'[\sigma/\alpha]$ is substitutible for β in $\tau'[\sigma/\alpha]$ for $i = 1, 2$, i.e. the substitution expressions $\tau'[\sigma/\alpha]\,[\tau_i'[\sigma/\alpha]/\beta]$ are admissible for $i = 1, 2$. It furthermore follows that σ is also substitutible for α in $\tau'[\tau_i'/\beta]$ for $i = 1, 2$, i.e. that the substitution expressions $(\tau'[\tau_1'/\beta])\,[\sigma/\alpha]$ are admissible for $i = 1, 2$. And due to Lemma 3.4.2, (3.18), $\tau'[\tau_i'/\beta] \equiv_{\mathrm{ren}} \tau[\tau_i/\beta]$ follows.

Under these circumstances the derivation $\mathcal{D}$ can be transformed into a derivation $\tilde{\mathcal{D}}$ of the form

$$\dfrac{\dfrac{\begin{array}{c}\mathcal{D}_{11}\\ \tau_1 = \tau[\tau_1/\beta]\end{array}}{\tau_1'[\sigma/\alpha] = \underbrace{(\tau'[\tau_1'/\beta])\,[\sigma/\alpha]}_{\underset{(\mathrm{I})}{\Leftrightarrow}\ \tau'[\sigma/\alpha]\,[\tau_1'[\sigma/\alpha]/\beta]}}\ \mathrm{S}._{(.)'[\sigma/\alpha]} \qquad \dfrac{\begin{array}{c}\mathcal{D}_{12}\\ \tau_2 = \tau[\tau_2/\beta]\end{array}}{\tau_2'[\sigma/\alpha] = \underbrace{(\tau'[\tau_2'/\beta])\,[\sigma/\alpha]}_{\underset{(\mathrm{II})}{\Leftrightarrow}\ \tau'[\sigma/\alpha]\,[\tau_2'[\sigma/\alpha]/\beta]}}\ \mathrm{S}._{(.)'[\sigma/\alpha]}}{\tau_1'[\sigma/\alpha] = \tau_2'[\sigma/\alpha]}\ \mathrm{UFP}$$

(here SUBST$_{(.)'[\sigma/\alpha]}$ has twice been abbreviated to S.$_{(.)'[\sigma/\alpha]}$ for typographical reasons) where for the purpose of establishing the equality equivalences (I) and (II) Lemma 3.3.11, (iii), has been used. The justification $\beta \downarrow \tau'[\sigma/\alpha]$ for the final UFP-application in $\tilde{\mathcal{D}}$ follows here from $\beta \downarrow \tau$ (which is the side condition on the application of UFP at the bottom of $\mathcal{D}$) and either of the assumptions $\beta \notin \mathrm{fv}(\sigma)$ or $\alpha \not\equiv \beta$ with the help of Lemma 7.1.2, (i) and (ii).

$\tilde{\mathcal{D}}$ can now effectively be transformed into an $\mathbf{AC}^{=}$-derivation $\mathcal{D}'$ without assumptions and with the same conclusion by using the induction hypothesis twice for the two subderivations of $\tilde{\mathcal{D}}$ ending in the two displayed applications of $\mathrm{SUBST}_{(.)'[\sigma/\alpha]}$ and by replacing these by $\mathbf{AC}^{=}$-derivations without assumptions and with the same respective conclusions.

This concludes the examination of all rules of $\mathbf{AC}^{=}$ for the induction step. Hence the proof by induction of the possibility to transform a derivation (7.17) (with the assumptions on the recursive types $\tau_1, \tau_2, \tau_1', \tau_2'$ and its subderivation $\mathcal{D}_1$ as above) has thus been completed.

(2) Here we will show the part of assertion (7.16) that refers to substitution rules of the family $\{\mathrm{SUBST}_{\tau[./\alpha]}\}_{\tau,\alpha}$ introduced in Definition 7.1.7, (ii). We will do so by proving the stronger assertion that these rules are in fact derivable in $\mathbf{AC}^{=}$ and can effectively be eliminated from all derivations in $\mathbf{AC}^{=}+$ $+\,\{\mathrm{SUBST}_{\tau[./\alpha]}\}_{\tau,\alpha}$.

We have to show the following part of (7.16) here: every derivation $\mathcal{D}$ of the form

$$\frac{\begin{array}{c}\mathcal{D}_1\\ \sigma_1 = \sigma_2\end{array}}{\tau[\sigma_1/\alpha] = \tau[\sigma_2/\alpha]}\,\mathrm{SUBST}_{\tau[./\alpha]} \tag{7.18}$$

where $\tau, \sigma_1, \sigma_2 \in \mu Tp$ and $\alpha \in TVar$ and where $\mathcal{D}_1$ is a derivation in $\mathbf{AC}^{=}$ without assumptions, can effectively be transformed into a derivation $\mathcal{D}'$ in $\mathbf{AC}^{=}$ without assumptions and with the same conclusion as $\mathcal{D}$. However, if rules $\mathrm{SUBST}_{\tau[./\alpha]}$ are actually derivable rules of $\mathbf{AC}^{=}$, and if, what is slightly more, for every application ι of $\mathrm{SUBST}_{\tau[./\alpha]}$ a derivation $\mathcal{D}^{(\iota)}_{\mathrm{mim}}$ in $\mathbf{AC}^{=}$ that mimics the derivation $\mathcal{D}_{(\iota)}$ corresponding to ι can effectively be found, then the assertion in the last sentence follows: this is because every derivation $\mathcal{D}$ as in (7.18) can then be transformed into the derivation

$$\begin{array}{c}\mathcal{D}_1\\ [\sigma_1 = \sigma_2]\\ \mathcal{D}^{(\iota)}_{\mathrm{mim}}\\ \tau[\sigma_1/\alpha] = \tau[\sigma_2/\alpha]\end{array} \tag{7.19}$$

in $\mathbf{AC}^{=}$, where $\mathcal{D}^{(\iota)}_{\mathrm{mim}}$ mimics the application of $\mathrm{SUBST}_{\tau[./\alpha]}$ at the bottom of $\mathcal{D}$.

Hence we are left with proving that substitution rules $\mathrm{SUBST}_{\tau[./\alpha]}$, where $\tau \in \mu Tp$ and $\alpha \in TVar$, are derivable in $\mathbf{AC}^{=}$ and that mimicking derivations $\mathcal{D}^{(\iota)}_{\mathrm{mim}}$ for applications ι of such rules can always be found effectively. We will prove this by induction on $|\tau|$.

Let $\alpha \in TVar$ be arbitrary. We will show by induction on $|\tau|$ that, for all

$\tau \in \mu Tp$ and for all derivations $\mathcal{D}_{(\iota)}$ in $\mathbf{AC}^=$+SUBST$_{\tau[./\alpha]}$ of the form

$$\frac{\sigma_1 = \sigma_2}{\tau[\sigma_1/\alpha] = \tau[\sigma_2/\alpha]}\ \mathrm{SUBST}_{\tau[./\alpha]} \tag{7.20}$$

with assumption $\sigma_1 = \sigma_2$, where $\sigma_1, \sigma_2 \in \mu Tp$, it is possible to find effectively a derivation $\mathcal{D}^{(\iota)}_{\mathrm{mim}}$ in $\mathbf{AC}^=$ that mimics $\mathcal{D}_{(\iota)}$, i.e. a derivation of the form

$$\begin{array}{c} [\sigma_1 = \sigma_2] \\ \mathcal{D}^{(\iota)}_{\mathrm{mim}} \\ \tau[\sigma_1/\alpha] = \tau[\sigma_2/\alpha] \end{array} \tag{7.21}$$

for which $\mathcal{D}^{(\iota)}_{\mathrm{mim}} \precsim \mathcal{D}_{(\iota)}$ holds, and hence $\mathsf{concl}(\mathcal{D}^{(\iota)}_{\mathrm{mim}}) = \mathsf{concl}(\mathcal{D}_{(\iota)})$ as well as $\mathrm{set}(\mathsf{assm}(\mathcal{D}^{(\iota)}_{\mathrm{mim}})) \subseteq \mathrm{set}(\mathsf{assm}(\mathcal{D}_{(\iota)})) = \{\sigma_1 = \sigma_2\}$.

First we consider the base case $|\tau| = 0$ of the induction.

If $\tau \equiv \bot$ or $\tau \equiv \top$ or $\tau \equiv \beta$ with $\beta \not\equiv \alpha$, then both of $\tau[\sigma_1/\alpha]$ and $\tau[\sigma_2/\alpha]$ denote τ and hence $\mathcal{D}^{(\iota)}_{\mathrm{mim}}$ can be chosen as the axiom $\tau = \tau$ belonging to the axiom scheme (REFL) of $\mathbf{AC}^=$. If $\tau \equiv \alpha$ then $\tau[\sigma_1/\alpha]$ denotes σ_1 and hence $\mathcal{D}^{(\iota)}_{\mathrm{mim}}$ can be chosen as the assumption $\sigma_1 = \sigma_2$.

Secondly, we consider the induction step in which $|\tau| \geq 1$ holds.

In the first subcase, we assume that, for some $\tau_1, \tau_2 \in \mu Tp$, $\tau \equiv \tau_1 \to \tau_2$ is the case, and that $\mathcal{D}_{(\iota)}$ is a derivation of the form (7.20) with some $\sigma_1, \sigma_2 \in \mu Tp$. We denote by $\mathcal{D}_{(\iota_1)}$ and $\mathcal{D}_{(\iota_2)}$ the two derivations of the form (7.20) with τ_1 and τ_2 in place of τ (admissibility of the substitution expressions $\tau_j[\sigma_i/\alpha]$, for $i, j \in \{1, 2\}$, in the conclusions of $\mathcal{D}_{(\iota_1)}$ and $\mathcal{D}_{(\iota_2)}$ follows from the implicit side-conditions of the admissibility of the substitution expressions $\tau[\sigma_i/\alpha]$, for $i \in \{1, 2\}$, occurring in the conclusion of the assumed derivation $\mathcal{D}_{(\iota)}$). Then by the induction hypothesis respective derivations $\mathcal{D}^{(\iota_1)}_{\mathrm{mim}}$ and $\mathcal{D}^{(\iota_2)}_{\mathrm{mim}}$ in $\mathbf{AC}^=$ with $\mathcal{D}^{(\iota_1)}_{\mathrm{mim}} \precsim \mathcal{D}_{(\iota_1)}$ and $\mathcal{D}^{(\iota_2)}_{\mathrm{mim}} \precsim \mathcal{D}_{(\iota_2)}$ can be produced effectively. It follows that the derivation

$$\frac{\begin{array}{c} [\sigma_1 = \sigma_2] \\ \mathcal{D}^{(\iota_1)}_{\mathrm{mim}} \\ \tau_1[\sigma_1/\alpha] = \tau_1[\sigma_2/\alpha] \end{array} \qquad \begin{array}{c} [\sigma_1 = \sigma_2] \\ \mathcal{D}^{(\iota_2)}_{\mathrm{mim}} \\ \tau_2[\sigma_1/\alpha] = \tau_2[\sigma_2/\alpha] \end{array}}{\tau[\sigma_1/\alpha] = \tau[\sigma_2/\alpha]}\ \mathrm{ARROW}$$

in $\mathbf{AC}^=$ mimics $\mathcal{D}_{(\iota)}$ and hence can be chosen as the desired derivation $\mathcal{D}^{(\iota)}_{\mathrm{mim}}$.

In the second subcase, we assume that $\tau \equiv \mu\beta.\,\tau_0$, for some $\beta \in TVar$ and $\tau_0 \in \mu Tp$, is the case, and that $\mathcal{D}_{(\iota)}$ is a derivation of the form (7.20) with some $\sigma_1, \sigma_2 \in \mu Tp$. If $\alpha \notin \mathrm{fv}(\tau)$ holds, then $\tau[\sigma_1/\alpha]$ and $\tau[\sigma_2/\alpha]$ both denote τ and hence $\mathcal{D}^{(\iota)}_{\mathrm{mim}}$ can be chosen as the axiom $\tau = \tau$ of $\mathbf{AC}^=$. Hence we can now assume that $\alpha \in \mathrm{fv}(\tau)$. It follows that $\alpha \not\equiv \beta$ and $\alpha \in \mathrm{fv}(\tau_0)$. We denote by $\mathcal{D}_{(\iota_0)}$ the derivation of the form (7.20) with τ_0 in place of τ (admissibility of the substitution expression $\tau_0[\sigma_i/\alpha]$, for $i \in \{1,2\}$, in the conclusion of $\mathcal{D}_{(\iota_0)}$ follows by Lemma 3.3.10, (i), due to the implicit side-condition of the admissibility of $\tau[\sigma_i/\alpha]$, for $i \in \{1,2\}$, in the conclusion of the assumed derivation $\mathcal{D}_{(\iota)}$). By the induction hypothesis there exists a derivation $\mathcal{D}^{(\iota_0)}_{\mathrm{mim}}$ in $\mathbf{AC}^{=}$ with $\mathcal{D}^{(\iota_0)}_{\mathrm{mim}} \precsim \mathcal{D}_{(\iota_0)}$. It follows that

$$\dfrac{\begin{array}{c}[\sigma_1 = \sigma_2]\\ \mathcal{D}^{(\iota_0)}_{\mathrm{mim}}\\ \tau_0[\sigma_1/\alpha] = \tau_0[\sigma_2/\alpha]\end{array}}{\underbrace{\mu\beta.\,\tau_0[\sigma_1/\alpha]}_{\underset{\text{(I)}}{\Longleftrightarrow}\,(\mu\beta.\,\tau_0)[\sigma_1/\alpha]} \;=\; \underbrace{\mu\beta.\,\tau_0[\sigma_2/\alpha]}_{\underset{\text{(II)}}{\Longleftrightarrow}\,(\mu\beta.\,\tau_0)[\sigma_2/\alpha]}}\ \mu\text{-COMPAT}$$

is a derivation in $\mathbf{AC}^{=}$ that mimics the derivation $\mathcal{D}_{(\iota)}$ and hence can be chosen as the derivation $\mathcal{D}^{(\iota)}_{\mathrm{mim}}$ in this case; the equality equivalences labeled by (I) and (II) follow from Lemma 3.3.10 due to $\alpha \not\equiv \beta$ and the admissibility of $(\mu\beta.\,\tau_0)\,[\sigma_1/\alpha]$ and $(\mu\beta.\,\tau_0)\,[\sigma_2/\alpha]$.

In this way we have carried out the induction step and hence have concluded the proof by induction of the derivability in $\mathbf{AC}^{=}$ of substitution rules of Definition 7.1.7, (ii), and of the possibility to find respective mimicking derivations effectively.

(3) Here we have to show that part of the assertion (7.16) which refers to substitution rules $R = \mathrm{SUBST}_{(.)'[./\alpha]}$ for some type variable $\alpha \in \mathit{TVar}$. We will see that this is an easy consequence of the parts (1) and (2) of this proof, and in particular, of the admissibility in $\mathbf{AC}^{=}$ of the substitution rules of kinds (i) and (ii) of Definition 7.1.7.

For showing this, we fix an arbitrary type variable α. Let $\mathcal{D}$ be an arbitrary derivation without assumptions that is of the form

$$\dfrac{\begin{array}{c}\mathcal{D}_1\\ \tau_1 = \tau_2\end{array} \qquad \begin{array}{c}\mathcal{D}_2\\ \sigma_1 = \sigma_2\end{array}}{\tau_1'[\sigma_1/\alpha] = \tau_2'[\sigma_2/\alpha]}\ \mathrm{SUBST}_{(.)'[./\alpha]} \tag{7.22}$$

where $\mathcal{D}_1$ and $\mathcal{D}_2$ are $\mathbf{AC}^{=}$-derivations and where τ_1' and τ_2' are variants of τ_1 and τ_2 respectively. Let τ_2'' be another variant of τ_2 such that both σ_1 and σ_2 are substitutible for α in τ_2''. Then the derivation $\tilde{\mathcal{D}}$

$$\dfrac{\dfrac{\dfrac{\begin{array}{c}\mathcal{D}_1\\ \tau_1 = \tau_2\end{array}}{\tau_1'[\sigma_1/\alpha] = \tau_2''[\sigma_1/\alpha]}\ \mathrm{SUBST}_{(.)'[\sigma_1/\alpha]} \qquad \dfrac{\begin{array}{c}\mathcal{D}_2\\ \sigma_1 = \sigma_2\end{array}}{\tau_2''[\sigma_1/\alpha] = \tau_2''[\sigma_2/\alpha]}\ \mathrm{SUBST}_{\tau_2''[./\alpha]}}{\tau_1'[\sigma_1/\alpha] = \tau_2''[\sigma_2/\alpha]}\ \mathrm{TRANS}}{\tau_1'[\sigma_1/\alpha] = \tau_2'[\sigma_2/\alpha]}\ \mathrm{REN}_r$$

is a derivation in the extension of the system $\mathbf{AC}^=$ by adding the rules $\mathrm{SUBST}_{(.)'[\sigma_1/\alpha]}$, $\mathrm{SUBST}_{\tau_2''[./\alpha]}$, and REN_r. Due to items (1) and (2) of this proof, the two subderivations of $\tilde{\mathcal{D}}$ that end in the premises of the displayed application of TRANS can be replaced by effectively found $\mathbf{AC}^=$-derivations $\mathcal{D}_1'$ and $\mathcal{D}_2'$ without assumptions and with the same respective conclusion. Then in the resulting derivation $\tilde{\mathcal{D}}'$ the application REN_r at its bottom can be replaced by an application of TRANS with an axiom (REN). In this way we have effectively eliminated the application of $\mathrm{SUBST}_{(.)'[./\alpha]}$ at the bottom of the assumed derivation $\mathcal{D}$ of the form (7.22), and we have reached, in an effective way, an $\mathbf{AC}^=$-derivation $\mathcal{D}'$ without assumptions and with the same conclusion $\tau_1'[\sigma_1/\alpha] = \tau_2'[\sigma_2/\alpha]$ as $\mathcal{D}$.

□

Remark 7.1.10. Assertion (ii) of Lemma 7.1.9, the proof of which was certainly not equally involved as that of assertion (i), is actually used by Amadio and Cardelli at various places in [AmCa93]. Notably, it is used there in example (2) of Section 5.1.2, "Derived Rules", on page 29, where it is shown that the formulas of the scheme of axioms $(\mu\mu-\mu)$ in Lemma 3.8.4 are in fact theorems of $\mathbf{AC}^=$ (in Example 5.1.8 we have demonstrated that the axioms of the scheme $(\mu\mu-\mu)$ are in fact theorems of the variant system $\mathbf{AC}^=_*$ of $\mathbf{AC}^=$). And furthermore, the admissibility of the rules $\mathrm{SUBST}_{\tau[./\alpha]}$ (for arbitrary τ and α) is also essentially used in the proof for Lemma 5.2.2, "A system of contractive equations has a unique solution", on page 30 in [AmCa93].

As an aside, we observe that the assertion of Lemma 7.1.9 stays correct if the substitution rules of kind (ii) in Definition 7.1.7 are replaced by the variant rules introduced in Remark 7.1.8.

Proposition 7.1.11. *For all $\tau \in \mu Tp$ and $\alpha \in TVar$ it holds that the substitution rule* $\mathrm{SUBST}_{(\tau)'[./\alpha]}$ *(defined in Remark 7.1.8) is derivable in* $\mathbf{AC}^=$*.*

Proof. For all $\tau \in \mu Tp$ and $\alpha \in TVar$, mimicking derivations in $\mathbf{AC}^=$ for applications of $\mathrm{SUBST}_{(\tau)'[./\alpha]}$ can be found from mimicking derivations of respective derivations of rules $\mathrm{SUBST}_{\tau'''[./\alpha]}$, for appropriate variants τ''' of τ, by appending one or two subsequent applications of $\mathrm{REN}_{l/r}$. □

Definition 7.1.12 (The variant systems $\mathbf{AC}^=_-$ and $\mathbf{AC}^=_{*-}$ of $\mathbf{AC}^=$). The (pure) Hilbert-style proof system $\mathbf{AC}^=_-$, a variant system of the system $\mathbf{AC}^=$ of Definition 5.1.1, is defined as follows: The *formulas* of $\mathbf{AC}^=_-$ are the equations between recursive types in *μTp–Eq*. The *axioms* of $\mathbf{AC}^=_-$ are all those, that belong to one of the axiom schemes (REFL), (REN) or (FOLD/UNFOLD) of Definition 5.1.1 together with those of the scheme

$$(\mu-\bot)' \; \overline{\mu\alpha\alpha_1\ldots\alpha_n.\,\alpha = \bot} \quad (\text{where } n \in \omega) \tag{7.23}$$

of the system $\mathbf{HB}^{=}$ (which axiom scheme properly contains the axioms $(\mu{-}\bot)$ of $\mathbf{AC}^{=}$). The *inference rules* of $\mathbf{AC}^{=}_{-}$ are the rules SYMM, TRANS, ARROW and UFP of $\mathbf{AC}^{=}$, i.e. $\mathbf{AC}^{=}_{-}$ contains all rules of $\mathbf{AC}^{=}$ except the rule μ-COMPAT.

Furthermore, we let $\mathbf{AC}^{=}_{*-}$ be the system that arises from the system $\mathbf{AC}^{=}_{*}$ of Definition 5.1.5 by removing, on the one hand, the axioms $(\mu{-}\bot)$ and the rule μ-COMPAT, and by adding, on the other hand, the axioms $(\mu{-}\bot)'$ of (7.23). ⊠

By this definition $\mathbf{AC}^{=}_{-}$ is an extension of the system $\mathbf{AC}^{=}{-}\mu$-COMPAT. In particular, every derivation in $\mathbf{AC}^{=}{-}\mu$-COMPAT is also a derivation in $\mathbf{AC}^{=}_{-}$. In the same way also $\mathbf{AC}^{=}_{*-}$ is an extension of the system $\mathbf{AC}^{=}_{*}{-}\mu$-COMPAT.

Proposition 7.1.13. *The systems* $\mathbf{AC}^{=}_{-}$ *and* $\mathbf{AC}^{=}_{*-}$ *are equivalent. What is more, every derivation* $\mathcal{D}$ *in* $\mathbf{AC}^{=}_{-}$ *can effectively be transformed (in an easy way) into a derivation* $\mathcal{D}_{*}$ *in* $\mathbf{AC}^{=}_{*-}$ *with the same conclusion and the same (if any) open assumptions as* $\mathcal{D}$*; and* vice versa.

Proof. This can be shown in the same way as the equivalence (and the manner to make this an "effective" assertion) of the systems $\mathbf{AC}^{=}$ and $\mathbf{AC}^{=}_{*}$ was shown in the proof of Proposition 5.1.6. □

For the main theorem of this section we will need the fact, stated by the following lemma, that not only are substitution rules of kind (i) in Definition 7.1.7 admissible in $\mathbf{AC}^{=}$ and in $\mathbf{AC}^{=}_{*}$ (as stated by Lemma 7.1.9, (ii)) but also in $\mathbf{AC}^{=}_{*}$ and in $\mathbf{AC}^{=}_{*-}$.

Lemma 7.1.14. *For all* $\sigma \in \mu Tp$ *and* $\alpha \in TVar$*, the rules* $\mathrm{SUBST}_{(.)'[\sigma/\alpha]}$ *are admissible in* $\mathbf{AC}^{=}_{-}$*. Moreover every derivation* $\mathcal{D}$ *in* $\mathbf{AC}^{=}_{-}+\{\,\mathrm{SUBST}_{(.)'[\sigma/\alpha]}\}_{\sigma,\alpha}$ *without assumptions can effectively be transformed into an* $\mathbf{AC}^{=}_{-}$*-derivation* $\mathcal{D}'$ *without assumptions and with the same conclusion.*

Proof. This can be shown by an analogous (in fact almost identical but shorter) proof than the one in item (1) of the proof of Lemma 7.1.9, (i). Only the following consequences of the difference between $\mathbf{AC}^{=}$ and $\mathbf{AC}^{=}_{-}$ have to be observed for an analogous proof by induction:

(a) For the base case of the induction: applications of $\mathrm{SUBST}(.)'[\sigma/\alpha]$ that immediately follow $\mathbf{AC}^{=}_{-}$-axioms of the scheme $(\mu - \bot)'$ amount to taking a variant on the left hand side of the equation in the axiom, since such axioms do not contain free variables and therefore substitutions of recursive types in them are to no effect; such applications can therefore always be replaced by applications of the transitivity rule TRANS with a variant axiom (REN).

(b) For the induction step:

  (a) Rules μ-COMPAT are not contained in $\mathbf{AC}^{=}_{-}$ and hence there is no need here to here treat upwards-permutation of rules $\mathrm{SUBST}_{(.)'[\sigma/\alpha]}$ over applications of μ-COMPAT-rules.

(b) The possibility to permute $\text{SUBST}_{(.)'[\sigma/\alpha]}$-rules upwards over $\mathbf{AC}^{=}$-rules other than μ-COMPAT does not depend on the presence of the rules μ-COMPAT in $\mathbf{AC}^{=}$ (and therefore all such permutations of $\mathbf{AC}^{=}_{-}$-rules can indeed be performed in $\mathbf{AC}^{=}_{-}$ in the same way done in the proof for item (i) of Lemma 7.1.9).

□

We are now able to prove the main theorem of this section. Informally, it states that the rule μ-COMPAT of $\mathbf{AC}^{=}$ and $\mathbf{AC}^{=}_{*}$ can be dispensed with in either of the systems $\mathbf{AC}^{=}_{-}$ and $\mathbf{AC}^{=}_{*-}$.

Theorem 7.1.15. *The rule μ-*COMPAT *is an admissible rule of $\mathbf{AC}^{=}_{-}$ and of $\mathbf{AC}^{=}_{*-}$. What is more, every derivation $\mathcal{D}$ of the form*

$$\dfrac{\begin{matrix}\mathcal{D}_1\\ \tau_1 = \tau_2\end{matrix}}{\mu\alpha.\,\tau_1 = \mu\alpha.\,\tau_2}\ \mu\text{-COMPAT}\ , \tag{7.24}$$

where $\mathcal{D}_1$ is an $\mathbf{AC}^{=}_{-}$-derivation without open assumptions, can effectively be transformed into an $\mathbf{AC}^{=}_{-}$-derivation $\mathcal{D}'$ with the same conclusion as $\mathcal{D}$ and without assumptions. And an analogous assertion holds with respect to derivations in the system $\mathbf{AC}^{=}_{*-}$.

Proof. Let $\mathcal{H}$ be one of the systems $\mathbf{AC}^{=}_{-}$ or $\mathbf{AC}^{=}_{*-}$. For showing that μ-COMPAT is admissible in $\mathcal{H}$, it suffices to prove that every derivation $\mathcal{D}$ of the form (7.24), where $\mathcal{D}_1$ denotes a derivation in $\mathcal{H}$ without assumptions, can effectively be transformed into a derivation $\mathcal{D}'$ in $\mathcal{H}$ with the same conclusion as $\mathcal{D}$ and without assumptions. In order to demonstrate this, we let $\mathcal{D}$ be an arbitrary derivation in $\mathcal{H}$ of the form (7.24) such that $\mathcal{D}_1$ is a derivation in $\mathcal{H}$ without assumptions. We distinguish two cases:

Case 1: $\alpha \not\downarrow \tau_1$ and $\alpha \not\downarrow \tau_2$.

Then by Lemma 7.1.1 it holds that $\tau_1 \equiv \mu\alpha_1 \ldots \alpha_{n_1}.\,\alpha$ and $\tau_2 \equiv \mu\tilde{\alpha}_1 \ldots \tilde{\alpha}_{n_2}.\,\alpha$ for some $n_1, n_2 \in \omega$ and variables $\alpha_1, \ldots, \alpha_{n_1}, \tilde{\alpha}_1 \ldots \tilde{\alpha}_{n_2} \not\equiv \alpha$. Then $\mu\alpha.\,\tau_1 = \bot$ and $\mu\alpha.\,\tau_2 = \bot$ are axioms of the scheme $(\mu - \bot)'$ of $\mathcal{H}$ and hence the derivation $\mathcal{D}'$

$$\dfrac{\dfrac{\scriptstyle(\mu\text{–}\bot)'}{\mu\alpha.\,\tau_1 = \bot} \qquad \dfrac{\dfrac{\scriptstyle(\mu\text{–}\bot)'}{\mu\alpha.\,\tau_2 = \bot}}{\bot = \mu\alpha.\,\tau_2}\ \text{SYMM}}{\mu\alpha.\,\tau_1 = \mu\alpha.\,\tau_2}\ \text{TRANS}$$

is a derivation in $\mathcal{H}$ with the same conclusion as $\mathcal{D}$ and without open assumptions.

Case 2: $\alpha \downarrow \tau_1$ or $\alpha \downarrow \tau_2$.

Suppose for once, that $\alpha \downarrow \tau_1$ holds (it will be obvious from the transformation described below that the case $\alpha \downarrow \tau_2$ can even be treated easier). Let τ_1' and τ_2' be variants of τ_1 and τ_2 respectively, such that $\mu\alpha.\,\tau_2$ (and then also $\mu\alpha.\,\tau_2'$) is substitutible for α in both τ_1' and τ_2'. Then $\alpha \downarrow \tau_1'$ holds as well (by Lemma 7.1.2, (i)), which justifies the application of CONTRACT in the following derivation $\mathcal{D}^{(1)}$:

$$\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{\overset{\mathcal{D}_1}{\tau_1 = \tau_2}}{\tau_2 = \tau_1}\;\text{SYMM}}{\tau_2'[\mu\alpha.\,\tau_2'/\alpha] = \tau_1'[\mu\alpha.\,\tau_2'/\alpha]}\;\text{SUBST}_{(.)'[\mu\alpha.\,\tau_2'/\alpha]}}{\mu\alpha.\,\tau_2' = \tau_1'[\mu\alpha.\,\tau_2'/\alpha]}\;\text{FOLD}_l}{\mu\alpha.\,\tau_2' = \mu\alpha.\,\tau_1'}\;\text{CONTRACT}}{\mu\alpha.\,\tau_2 = \mu\alpha.\,\tau_1}\;\text{REN}_l,\ \text{REN}_r}{\mu\alpha.\,\tau_1 = \mu\alpha.\,\tau_2}\;\text{SYMM}$$

The subderivation $\mathcal{D}^{(1)}_{1+}$ of $\mathcal{D}^{(1)}$, whose last rule application is the displayed application of $\text{SUBST}_{(.)'[\mu\alpha.\,\tau_2'/\alpha]}$ in $\mathcal{D}^{(1)}$, can—due to Lemma 7.1.14—effectively be transformed into an $\mathbf{AC}^{=}_{-}$-derivation $\mathcal{D}^{(2)}_{1+}$ with the same conclusion and without assumptions. Let the result of replacing $\mathcal{D}^{(1)}_{1+}$ in $\mathcal{D}^{(1)}$ by $\mathcal{D}^{(2)}_{1+}$ be $\mathcal{D}^{(2)}$. In the case $\mathcal{H} = \mathbf{AC}^{=}_{*-}$ the derivation $\mathcal{D}^{(2)}$ can be directly transformed into a derivation in $\mathcal{H}$ by eliminating the two displayed applications of $\text{REN}_{l/r}$, and the application of FOLD_l. In the case $\mathcal{H} = \mathbf{AC}^{=}_{-}$ additionally the single application of CONTRACT has to be removed from $\mathcal{D}^{(2)}$ (this can be done as shown in the proof of Proposition 5.1.6). In both cases the result effectively reached is a derivation $\mathcal{D}'$ in $\mathcal{H}$ with the same conclusion as $\mathcal{D}$ and without assumptions.

In both cases a derivation $\mathcal{D}'$ in $\mathcal{H}$ with the same conclusion $\mu\alpha.\,\tau_1 = \mu\alpha.\,\tau_2$ as $\mathcal{D}$ and without assumptions has effectively been found.

□

Remark 7.1.16. The proof of Theorem 7.1.15 yields some additional information: if $\mathbf{AC}^{=}_{-}$ or $\mathbf{AC}^{=}_{*-}$ is enriched by the substitution rules $\text{SUBST}_{(.)'[\sigma/\alpha]}$ (for all $\alpha \in \mathit{TVar}$ and $\sigma \in \mu\mathit{Tp}$), then μ-COMPAT is even a derivable rule. That is, μ-COMPAT is a derivable rule in each of the systems $\mathbf{AC}^{=}_{-}+\{\text{SUBST}_{(.)'[\sigma/\alpha]}\}_{\sigma,\alpha}$ and $\mathbf{AC}^{=}_{*-}+\{\text{SUBST}_{(.)'[\sigma/\alpha]}\}_{\sigma,\alpha}$.

Corollary 7.1.17. *The axiom systems* $\mathbf{AC}^{=}$, $\mathbf{AC}^{=}_{*}$, $\mathbf{AC}^{=}_{-}$ *and* $\mathbf{AC}^{=}_{*-}$ *are equivalent, i.e. all of these systems possess the same theorems. Furthermore every derivation* $\mathcal{D}$ *in either of these systems can effectively be transformed into a derivation* $\mathcal{D}'$ *in one of the other systems such that* $\mathcal{D}'$ *has the same conclusion and the same assumptions as* $\mathcal{D}$.

Proof. In view of Proposition 5.1.6 and of Proposition 7.1.13 it suffices to show the equivalence of $\mathbf{AC}^{=}$ and $\mathbf{AC}^{=}_{-}$ and the existence of effective transformations between these two systems.

To show on the one hand that every $\mathbf{AC}^{=}_{-}$-derivation can be transformed into an $\mathbf{AC}^{=}$-derivation with the same conclusion, it suffices to give derivations for every axiom of the scheme $(\mu-\bot)'$ of $\mathbf{AC}^{=}$ (because all other axioms of $\mathbf{AC}^{=}_{-}$ as well as all of its rules are also, respectively, axioms and rules of $\mathbf{AC}^{=}$). All axioms of the scheme $(\mu-\bot)'$ allow in fact derivations from axioms $(\mu-\bot)$ and (FOLD/UNFOLD) with rules TRANS and μ -COMPAT. Here only an example shall be presented: the axiom $\mu\alpha\alpha_2\alpha\alpha_1.\,\alpha$ (where $\alpha_1, \alpha_2 \not\equiv \alpha$) of $\mathbf{AC}^{=}_{-}$. This equation admits the derivation

$$\dfrac{\dfrac{\text{(FOLD/UNFOLD)}}{\mu\alpha\alpha_2\alpha\alpha_1.\,\alpha = \mu\alpha_2\alpha\alpha_1.\,\alpha} \qquad \dfrac{\dfrac{\text{(FOLD/UNFOLD)}}{\mu\alpha_2\alpha\alpha_1.\,\alpha = \mu\alpha\alpha_1.\,\alpha} \qquad \mathcal{D}_0}{\mu\alpha_2\alpha\alpha_1.\,\alpha = \bot}\,\text{TRANS}}{\mu\alpha\alpha_2\alpha\alpha_1.\,\alpha = \bot}\,\text{TRANS}$$

in $\mathbf{AC}^{=}$, where $\mathcal{D}_0$ is the derivation

$$\dfrac{\dfrac{\dfrac{\text{(FOLD/UNFOLD)}}{\mu\alpha_1.\,\alpha = \alpha}}{\mu\alpha\alpha_1.\,\alpha = \mu\alpha.\,\alpha}\,\mu\text{-COMPAT} \qquad \dfrac{(\mu-\bot)}{\mu\alpha.\,\alpha = \bot}}{\mu\alpha\alpha_1.\,\alpha = \bot}\,\text{TRANS}$$

To show on the other hand that every $\mathbf{AC}^{=}$-derivation can effectively be transformed into an $\mathbf{AC}^{=}_{-}$-derivation with the same conclusion, it suffices to prove (since derivations in $\mathbf{AC}^{=}-\mu$-COMPAT are derivations in $\mathbf{AC}^{=}_{-}$) that applications of μ-COMPAT can effectively be eliminated from an arbitrary $\mathbf{AC}^{=}$-derivation $\mathcal{D}$. But this follows clearly from the theorem by induction on the number of applications of μ-COMPAT in an $\mathbf{AC}^{=}$-derivation $\mathcal{D}$, where in the induction step always topmost occurrences of μ-COMPAT are considered and removed.

□

Remark 7.1.18. By a similar argument as used in the above proof for the derivability of the axioms $(\mu-\bot)$ in $\mathbf{AC}^{=}$, it can be shown that the axioms of the scheme $(\mu-\bot)'$ in $\mathbf{AC}^{=}_{-}$ can actually be derived from ones belonging to the proper subscheme

$$(\mu-\bot)'' \quad \overline{\mu\alpha\alpha_1\ldots\alpha_n.\,\alpha = \bot} \qquad (\text{where } n \in \omega \text{ and } \alpha \not\equiv \alpha_1, \ldots, \alpha_n\,) \tag{7.25}$$

of $(\mu-\bot)'$ by derivations using additionally only axioms (FOLD/UNFOLD) and applications of TRANS. Consequently, it had been possible to take up the axiom scheme $(\mu-\bot)''$ of (7.25) instead of the axiom scheme $(\mu-\bot)'$ from (7.23) into the definition of the systems $\mathbf{AC}^{=}_{-}$ and $\mathbf{AC}^{=}_{*-}$ in Definition 7.1.12 and equivalent systems would have resulted respectively. We have chosen not to do so in the aim of avoiding the obligation to carry along the inessential restriction $\alpha \not\equiv \alpha_1, \ldots, \alpha_n$ on the axioms of the form (7.23).

The following theorem states that, with the single exception of the statement concerning derivability of rules $\mathrm{SUBST}_{\tau[./\alpha]}$, for all $\tau \in \mu Tp$ and $\alpha \in TVar$, the assertion of Lemma 7.1.9 stays true if the system $\mathbf{AC}^{=}$ is replaced everywhere by the system $\mathbf{AC}^{=}_{-}$. We will show this theorem by appropriately applying the proof-theoretic transformations developed in the proofs of this section.

Theorem 7.1.19. *All of the substitution rules from Definition 7.1.7 are admissible in* $\mathbf{AC}^{=}_{-}$ *and can effectively be eliminated from derivations in the extension of* $\mathbf{AC}^{=}_{-}$ *by adding these additional rules.*

Proof. It suffices to show that every derivation without assumptions in the extension of $\mathbf{AC}^{=}_{-}$ by the substitution rules from Definition 7.1.7 can effectively be transformed into a derivation in $\mathbf{AC}^{=}_{-}$.

Let an arbitrary derivation $\mathcal{D}$ without assumptions in the extension of $\mathbf{AC}^{=}_{-}$ by the substitution rules from Definition 7.1.7 be given. The derivation $\mathcal{D}$ can be effectively transformed into a derivation $\mathcal{D}'$ in $\mathbf{AC}^{=}_{-}$ without assumptions and with the same conclusion as $\mathcal{D}$ be performing the following four steps:

(1) *Elimination of substitution rules from Definition 7.1.7, (iii):* Using the transformation described in item (3) of the proof of Lemma 7.1.9, it is possible to eliminate from $\mathcal{D}$ all applications of the substitution rule $\mathrm{SUBST}_{(.)'[./\alpha]}$, for arbitrary $\alpha \in TVar$. Hereby this transformation is applied successively to arbitrarily chosen occurrences of applications of $\mathrm{SUBST}_{(.)'[./\alpha]}$, thereby eliminating the respective application and introducing no new applications of such rules (but introducing substitution rules of kind (i) and (ii) in Definition 7.1.9 instead), until no applications of rules $\mathrm{SUBST}_{(.)'[./\alpha]}$, for some $\alpha \in TVar$ are left. The result $\mathcal{D}^{(1)}$ of this elimination process is then a derivation in the extension of $\mathbf{AC}^{=}_{-}$ by adding the substitution rules of kinds (i) and (ii) in Definition 7.1.7.

(2) *Elimination of substitution rules from Definition 7.1.7, (ii):* Using the transformation described in item (2) of the proof of Lemma 7.1.9, it is possible to eliminate from $\mathcal{D}^{(1)}$ all applications of substitution rules $\mathrm{SUBST}_{\tau[./\alpha]}$, for arbitrary $\tau \in \mu Tp$ and $\alpha \in TVar$. In each such elimination-step of an arbitrarily chosen application of a rule $\mathrm{SUBST}_{\tau[./\alpha]}$ no new occurrences of such rules are introduced but possibly the rule μ-COMPAT (which is not present in $\mathbf{AC}^{=}_{-}$) is used. The result $\mathcal{D}^{(2)}$ is then a derivation without assumptions and with the same conclusion as $\mathcal{D}$ in the extension of $\mathbf{AC}^{=}_{-}$ by adding the rule μ-COMPAT as well as substitution rules of kind (i) in Definition 7.1.7.

(3) *Elimination of* μ-COMPAT*:* All applications of the rule μ-COMPAT in $\mathcal{D}^{(2)}$ can successively be eliminated in an effective way by using the transformation from the proof of the Theorem 7.1.15. Each such elimination step (applied first to $\mathcal{D}^{(2)}$ and then to the intermediate-results of derivations still containing applications of μ-COMPAT) can be done in such a way that no new application of μ-COMPAT is introduced, but only axioms of $\mathbf{AC}^{=}$, applications of rules of $\mathbf{AC}^{=}_{-}$, and applications of substitution rules a $\mathrm{SUBST}_{(.)'[\sigma/\alpha]}$, for some σ, α.

The result of these successive and stepwise eliminations of all applications of μ-COMPAT in $\mathcal{D}^{(2)}$ is a derivation $\mathcal{D}^{(3)}$ without assumptions and with the same conclusion as $\mathcal{D}$ in the extension of $\mathbf{AC}^{=}$ with substitution rules of kind (i) in Definition 7.1.7.

(4) *Elimination of substitution rules from Definition 7.1.7, (i):* By Lemma 7.1.14 it follows now that $\mathcal{D}^{(3)}$ can effectively be transformed by the elimination of applications of $\text{SUBST}_{(.)'[\sigma/\alpha]}$ into an $\mathbf{AC}^{=}_{-}$-derivation $\mathcal{D}'$ without assumptions and with the same conclusion as $\mathcal{D}$.

□

We conclude this section with two propositions that complement the 'positive' statements of the substitution rules and of the rule μ-COMPAT being admissible in $\mathbf{AC}^{=}_{-}$ and in $\mathbf{AC}^{=}_{*-}$ by 'negative' statements concerning derivability in these systems of μ-COMPAT and of substitution rules of kind (i) in Definition 7.1.7.

Proposition 7.1.20. *The rule* μ-COMPAT *is neither derivable in* $\mathbf{AC}^{=}_{-}$ *nor in* $\mathbf{AC}^{=}_{*-}$.

Proof. It suffices to prove that μ-COMPAT is not a derivable rule in $\mathbf{AC}^{=}_{-}$ because, as a consequence of Proposition 7.1.13, for every derivation $\mathcal{D}$ in $\mathbf{AC}^{=}_{-}$ that mimics an application ι of μ-COMPAT there exists a derivation $\mathcal{D}_*$ in $\mathbf{AC}^{=}_{*-}$ that also mimics ι. We will give two auxiliary assertions in item (a) below and will conclude the proof in item (b).

(a) We notice first the consequence

$$\Big[\ \vdash_{\mathbf{AC}^{=}_{-}} \tau = \sigma \quad \Rightarrow \quad \mathcal{L}'(\tau) = \mathcal{L}'(\sigma)\ \Big] \quad \text{(for all } \tau, \sigma \in \mu Tp) \tag{7.26}$$

of Lemma 5.3.1, (iii). In an alternative way, (7.26) follows from the soundness of $\mathbf{AC}^{=}_{-}$ with respect to $=_\mu$ (because $\mathbf{AC}^{=}_{-}$ and $\mathbf{AC}^{=}$ are equivalent, the soundness of $\mathbf{AC}^{=}_{-}$ with respect to $=_\mu$ is entailed by the soundness of $\mathbf{AC}^{=}$ with respect to $=_\mu$).

Secondly, also the statement

$$\left[\ \begin{array}{c} [\alpha = \beta] \\ \mathcal{D} \\ \tau = \sigma \end{array} \quad \begin{array}{l} \text{is a derivation in } \mathbf{AC}^{=}_{-} \text{ with at least one assm.} \\ \text{of the form } \alpha = \beta \text{ and with no other assms.} \end{array} \quad \Longrightarrow \right.$$

$$\left. \Longrightarrow \quad \mathcal{L}'(\tau), \mathcal{L}'(\sigma) \in \{\alpha, \beta, \rightarrow\}\ \right] \quad \text{(for all } \tau, \sigma \in \mu Tp) \tag{7.27}$$

can be shown in a quite straightforward way by induction on the depth $|\mathcal{D}|$ of a derivation $\mathcal{D}$ in $\mathbf{AC}^{=}_{-}$ with at least one occurrence of a assumption of the form $\alpha = \beta$ and without other assumptions. In the induction step the assertion (7.26) is used for settling one subcase of the case with an application of TRANS at the bottom of $\mathcal{D}$.

(b) μ-COMPAT is not a derivable rule in $\mathbf{AC}^{=}_{-}$:

Suppose, to the contrary, that μ-COMPAT is a derivable rule in $\mathbf{AC}^{=}_{-}$ and let α and β be type variables such that $\alpha \not\equiv \beta$. Then for the application of μ-COMPAT of the form

$$\frac{\alpha = \beta}{\mu\alpha.\,\alpha = \mu\alpha.\,\beta}\ \mu\text{-COMPAT}$$

there exists a derivation $\mathcal{D}$ in $\mathbf{AC}^{=}_{-}$ that has conclusion $\mu\alpha.\,\alpha = \mu\alpha.\,\beta$ and that does not contain assumptions other than possibly such of the form $\alpha = \beta$. If $\mathcal{D}$ does not contain assumptions at all, then its conclusion $\mu\alpha.\,\alpha = \mu\alpha.\,\beta$ is a theorem of $\mathbf{AC}^{=}_{-}$; but this contradicts (7.26) because of

$$\mathcal{L}'(\mu\alpha.\,\alpha) = \bot \quad \text{and} \quad \mathcal{L}'(\mu\alpha.\,\beta) = \beta\ . \tag{7.28}$$

However, if $\mathcal{D}$ contains assumptions of the form $\alpha = \beta$, then a contradiction with (7.27) arises, again due to (7.28). Thus our assumption that μ-COMPAT is derivable in $\mathbf{AC}^{=}_{-}$ cannot be sustained.

□

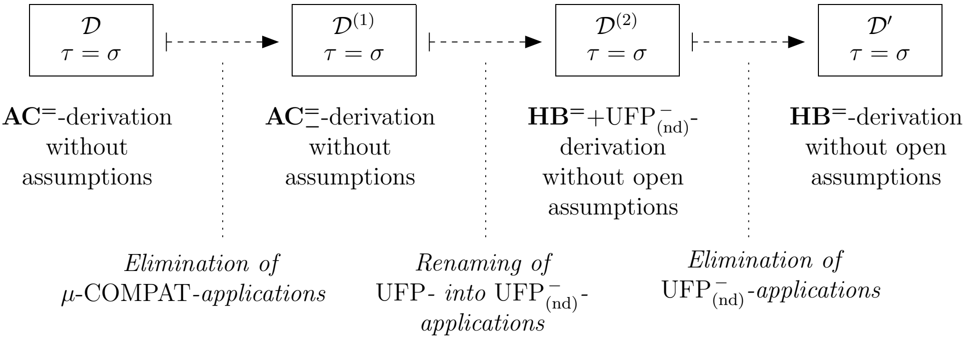
Proposition 7.1.21. *Not all of the substitution rules* $\text{SUBST}_{(\cdot)'[\sigma/\alpha]}$*, for* $\sigma \in \mu Tp$ *and* $\alpha \in TVar$*, are derivable in* $\mathbf{AC}^{=}$*. And the same assertion holds with respect to* $\mathbf{AC}^{=}_{*}$*.*

Proof. Close inspection of the proof of Lemma 7.1.15 shows that μ-COMPAT would be derivable in both $\mathbf{AC}^{=}$ and $\mathbf{AC}^{=}_{*}$ if all substitution rules $\text{SUBST}_{(\cdot)'[\sigma/\alpha]}$, for $\sigma \in \mu Tp$ and $\alpha \in TVar$, were derivable in both of these systems (we have observed a related statement in Remark 7.1.16). However, Proposition 7.1.20 tells us that μ-COMPAT is neither derivable in $\mathbf{AC}^{=}$ nor in $\mathbf{AC}^{=}_{*}$. Therefore the statement of the proposition follows. □

7.2 A Transformation of $\mathbf{AC}^{=}$-Derivations via $\mathbf{AC}^{=}_{-}$-Derivations into $\mathbf{HB}^{=}$-Derivations

In this section we describe and justify an effective proof-theoretic transformation from derivations in the proof system $\mathbf{AC}^{=}$ of Amadio and Cardelli into derivations in the proof system $\mathbf{HB}^{=}$ of Brandt and Henglein. This transformation produces, starting from a derivation $\mathcal{D}$ in $\mathbf{AC}^{=}$ without assumptions, a derivation $\mathcal{D}'$ in $\mathbf{HB}^{=}$ without open assumptions and with the same conclusion as $\mathcal{D}$. It proceeds in three main steps, the first of which will be based on preparatory work done in the previous section, while the second and the third will be developed in this section. In the first of these steps, a given derivation $\mathcal{D}$ in $\mathbf{AC}^{=}$ without assumptions is transformed into a derivation $\mathcal{D}^{(1)}$ in the variant system $\mathbf{AC}^{=}_{-}$ of $\mathbf{AC}^{=}$ such that $\mathcal{D}^{(1)}$ has the same conclusion as $\mathcal{D}$ and does not contain assumptions. And in the second and

Figure 7.1: Illustration of the three main steps in the transformation developed in this section from derivations in $\mathbf{AC}^{=}$ without assumptions into mimicking derivations in $\mathbf{HB}^{=}$.



$\mathcal{D}$ $\tau = \sigma$
$\mathcal{D}^{(1)}$ $\tau = \sigma$
$\mathcal{D}^{(2)}$ $\tau = \sigma$
$\mathcal{D}'$ $\tau = \sigma$
$\mathbf{AC}^{=}$-derivation without assumptions
$\mathbf{AC}^{=}_{-}$-derivation without assumptions
$\mathbf{HB}^{=}$+UFP$^{-}_{(\text{nd})}$-derivation without open assumptions
$\mathbf{HB}^{=}$-derivation without open assumptions
Elimination of μ-COMPAT-*applications*
Renaming of UFP- *into* UFP$^{-}_{(\text{nd})}$-*applications*
Elimination of UFP$^{-}_{(\text{nd})}$-*applications*

third steps, the derivation $\mathcal{D}^{(1)}$ in $\mathbf{AC}^{=}_{-}$ is then transformed into a derivation $\mathcal{D}'$ in $\mathbf{HB}^{=}$ with the same conclusion as $\mathcal{D}$ and without open assumption classes.

Now we are going to outline, in some more detail, the three main steps of the transformation in their application to a derivation in $\mathbf{AC}^{=}$. For this purpose we let an arbitrary derivation $\mathcal{D}$ in $\mathbf{AC}^{=}$ be given. We refer to Figure 7.1 for an illustration of this transformation, and start by sketching its first step:

(1) μ-COMPAT-*elimination Step*: The $\mathbf{AC}^{=}$-derivation $\mathcal{D}$ without assumptions is transformed, by an effective process of μ-COMPAT-elimination, into a mimicking derivation $\mathcal{D}^{(1)}$ in $\mathbf{AC}^{=}_{-}$, i.e. a derivation without assumptions and with the same conclusion as $\mathcal{D}$.

For the existence of such a process of μ-COMPAT-elimination we will be able to refer to Corollary 7.1.17; a particular procedure of this kind underlies the proof of this corollary, and can be extracted from there and from the proofs of some other statements in Section 7.1, to which the proof of Corollary 7.1.17 refers to (most importantly, from the proof of Theorem 7.1.15, for effective μ-COMPAT-elimination, and from the proof of Lemma 7.1.9, for effective elimination of substitution rules).

The second and the third steps of the transformation will be based on the following observation: the systems $\mathbf{AC}^{=}_{-}$ and $\mathbf{HB}^{=}$ have the same axioms and they differ only in the rule UFP, which is a rule of $\mathbf{AC}^{=}$, but not of $\mathbf{HB}^{=}$, and in the rule ARROW/FIX, which is a rule $\mathbf{HB}^{=}$, but not of $\mathbf{AC}^{=}$; however, we have ignored in this sentence that $\mathbf{AC}^{=}_{-}$ is a pure Hilbert-system, whereas $\mathbf{HB}^{=}$ is a natural-deduction system. Still, it follows that if it is possible to 'mimic' arbitrary applications of UFP in $\mathbf{HB}^{=}$, then every derivation in $\mathbf{AC}^{=}_{-}$ can be transformed, by successively replacing applications of UFP by appropriate mimicking derivations (more precisely, by mimicking derivation-contexts), into a derivation in $\mathbf{HB}^{=}$ without open assumptions and with the same conclusion. These considerations form the basis for the second and the third step, which are are outlined below:

(2) *Renaming Step of* UFP- *into* $\mathrm{UFP}^{-}_{(\mathrm{nd})}$*-applications*: The derivation $\mathcal{D}^{(1)}$ in $\mathbf{AC}^{=}_{-}$ without assumptions is altered into a derivation $\mathcal{D}^{(2)}$ without open assumptions in an extension of $\mathbf{HB}^{=}$ with an appropriate natural-deduction-system counterpart $\mathrm{UFP}^{-}_{(\mathrm{nd})}$ of the rule UFP in $\mathbf{AC}^{=}$. This step will always consist purely of renaming applications of UFP in $\mathcal{D}^{(1)}$ into applications of $\mathrm{UFP}^{-}_{(\mathrm{nd})}$ in $\mathcal{D}^{(2)}$. This modification of $\mathcal{D}^{(1)}$ does not affect the conclusion of the derivation, and hence $\mathcal{D}^{(2)}$ has the same conclusion as $\mathcal{D}^{(1)}$, and therefore also the same conclusion as $\mathcal{D}$.

(3) $\mathrm{UFP}^{-}_{(\mathrm{nd})}$*-elimination Step*: In this step all applications of $\mathrm{UFP}^{-}_{(\mathrm{nd})}$ are effectively eliminated from the derivation $\mathcal{D}^{(2)}$ in $\mathbf{HB}^{=}+\mathrm{UFP}^{-}_{(\mathrm{nd})}$ with the result of a derivation $\mathcal{D}'$ in $\mathbf{HB}^{=}$ without open assumptions and with the same conclusion as $\mathcal{D}^{(2)}$, and thereby also with the same conclusion as $\mathcal{D}$.

The second step of the transformation will later be justified by Proposition 7.2.2 (i). And the third step will be based upon Lemma 7.2.3, (ii), for which we will actually show that the mentioned natural-deduction-system version $\mathrm{UFP}^{-}_{(\mathrm{nd})}$ of the rule UFP is derivable in $\mathbf{HB}^{=}$, and hence that every application of $\mathrm{UFP}^{-}_{(\mathrm{nd})}$ in $\mathbf{HB}^{=}+\mathrm{UFP}^{-}_{(\mathrm{nd})}$ can be mimicked by an appropriate derivation context in $\mathbf{HB}^{=}$ without open assumptions.

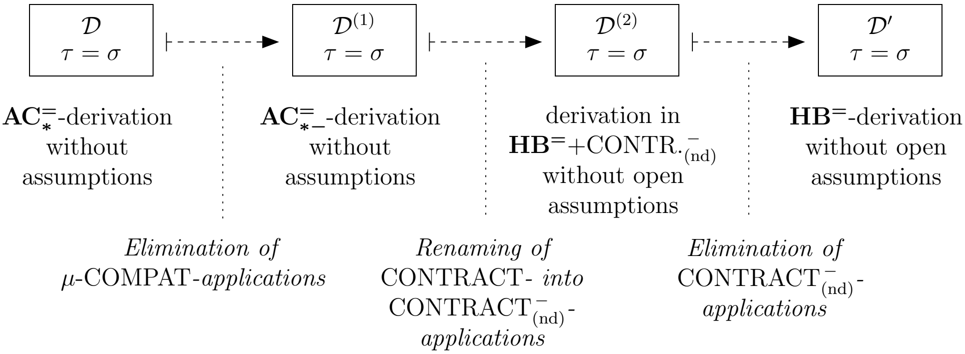
Contrasting with the elimination of μ-COMPAT-applications from an $\mathbf{AC}^{=}$-derivation without assumptions, which needs a somewhat complicated procedure related to the fact that μ-COMPAT is only admissible, but not derivable in $\mathbf{AC}^{=}_{-}$, the elimination of $\mathrm{UFP}^{-}_{(\mathrm{nd})}$-applications from a derivation $\mathcal{D}^{(2)}$ in $\mathbf{HB}^{=}+\mathrm{UFP}^{-}_{(\mathrm{nd})}$, as this takes place in the third step of the transformation, is a comparably easier process due to the fact that $\mathrm{UFP}^{-}_{(\mathrm{nd})}$ will be recognized to be a derivable rule in $\mathbf{HB}^{=}$. This implies namely that it is possible to directly "translate" applications ι of $\mathrm{UFP}^{-}_{(\mathrm{nd})}$ into mimicking derivation-contexts $\mathcal{DC}^{(\iota)}_{\mathrm{mim}}$ in $\mathbf{HB}^{=}$, with the effect that an application $\tilde{\iota}$ of $\mathrm{UFP}^{-}_{(\mathrm{nd})}$ at the bottom of a derivation $\tilde{\mathcal{D}}$ in $\mathbf{HB}^{=}+\mathrm{UFP}^{-}_{(\mathrm{nd})}$ like

$$\frac{\begin{array}{c}\tilde{\mathcal{D}}_1\\ \tau_1 = \tau[\tau_1/\alpha]\end{array} \qquad \begin{array}{c}\tilde{\mathcal{D}}_2\\ \tau_2 = \tau[\tau_2/\alpha]\end{array}}{\tau_1 = \tau_2}\,\mathrm{UFP}^{-}_{(\mathrm{nd})} \tag{7.29}$$

can be eliminated by replacing $\tilde{\mathcal{D}}$ with the derivation $\tilde{\mathcal{D}}'$ in $\mathbf{HB}^{=}+\mathrm{UFP}^{-}_{(\mathrm{nd})}$ of the form

$$\begin{array}{c}\begin{array}{cc}\tilde{\mathcal{D}}_1 & \tilde{\mathcal{D}}_2\\ [\,\tau_1 = \tau[\tau_1/\alpha]\,]_1 & [\,\tau_2 = \tau[\tau_2/\alpha]\,]_2\end{array}\\ \mathcal{DC}^{(\tilde{\iota})}_{\mathrm{mim}}\\ \tau_1 = \tau_2\end{array} \tag{7.30}$$

Figure 7.2: Illustration of the three main steps in the transformation developed in this section from derivations in $\mathbf{AC}^=_*$ without assumptions into mimicking derivations in $\mathbf{HB}^=$.


$\mathcal{D}$ $\tau = \sigma$
$\mathcal{D}^{(1)}$ $\tau = \sigma$
$\mathcal{D}^{(2)}$ $\tau = \sigma$
$\mathcal{D}'$ $\tau = \sigma$
$\mathbf{AC}^=_*$-derivation without assumptions
$\mathbf{AC}^=_{*-}$-derivation without assumptions
derivation in $\mathbf{HB}^=$+CONTR.$^-_{(\text{nd})}$ without open assumptions
$\mathbf{HB}^=$-derivation without open assumptions
Elimination of μ-COMPAT-*applications*
Renaming of CONTRACT- *into* CONTRACT$^-_{(\text{nd})}$-*applications*
Elimination of CONTRACT$^-_{(\text{nd})}$-*applications*

where $\mathcal{DC}^{(\tilde{\iota})}_{\text{mim}} \in \mathcal{D}er\mathcal{C}txt_2(\mathbf{HB}^=)$ is a mimicking derivation context for $\tilde{\iota}$ in $\mathbf{HB}^=$. However, the number of UFP$^-_{(\text{nd})}$-applications may actually increase during such an elimination step if both context-holes $[\,]_1$ and $[\,]_2$ occur in $\mathcal{DC}^{(\iota)}_{\text{mim}}$ and if, for some $i \in \{1,2\}$, the derivation $\mathcal{D}_i$ contains UFP$^-_{(\text{nd})}$-applications and the derivation context $\mathcal{DC}^{(\tilde{\iota})}_{\text{mim}}$ contains more than one occurrence of the context-hole $[\,]_i$. But the elimination of topmost occurrences of UFP$^-_{(\text{nd})}$ in such a transformation step does actually decrease the number of UFP$^-_{(\text{nd})}$-applications in a derivation in $\mathbf{HB}^=$; more precisely, if, for the derivation $\tilde{\mathcal{D}}$ in (7.29), the subderivations $\tilde{\mathcal{D}}_1$ and $\tilde{\mathcal{D}}_2$ are actually $\mathbf{HB}^=$-derivations, then the respective transformed derivation $\tilde{\mathcal{D}}'$ is also a derivation in $\mathbf{HB}^=$ (we will show that $\mathcal{DC}^{(\iota)}_{\text{mim}}$ can be chosen such that $\tilde{\mathcal{D}}$ and $\tilde{\mathcal{D}}'$ have the same open assumption classes).

We will also describe an analogous transformation from derivations $\mathcal{D}$ without assumptions in the variant system $\mathbf{AC}^=_*$ of $\mathbf{AC}^=$ into derivations $\mathcal{D}'$ in $\mathbf{HB}^=$ without open assumptions and with the same conclusion as $\mathcal{D}$. This transformation is diagrammatically pictured in Figure 7.2, which is analogous to Figure 7.1, the schematic illustration we have given above for the transformation between $\mathbf{AC}^=$- and $\mathbf{HB}^=$-derivations. Here, a suitable natural-deduction-system version CONTRACT$^-_{(\text{nd})}$ of the $\mathbf{AC}^=_*$-rule CONTRACT comes into play. In the first step, a derivation $\mathcal{D}$ in $\mathbf{AC}^=_*$ without assumptions is transformed into a derivation $\mathcal{D}^{(1)}$ in $\mathbf{AC}^=_{*-}$ such that $\mathcal{D}^{(1)}$ does not contain assumptions and has the same conclusion as $\mathcal{D}$. Then in the second step, each occurrence of an application of CONTRACT in $\mathcal{D}^{(1)}$ is merely renamed into an occurrence of CONTRACT$^-_{(\text{nd})}$ with the result of a derivation $\mathcal{D}^{(2)}$ in $\mathbf{HB}^=$+CONTRACT$^-_{(\text{nd})}$ without open assumptions that has the same conclusion as $\mathcal{D}$. Finally in the third step, $\mathcal{D}^{(2)}$ is transformed into a derivation $\mathcal{D}'$ in $\mathbf{HB}^{=}$ without open assumptions and with the same conclusion as $\mathcal{D}$ by a process of effective $\text{CONTRACT}^{-}_{(\text{nd})}$-elimination.

We are going to present the proof for the existence of the effective transformation from arbitrary $\mathbf{AC}^{=}$-derivations without assumptions into $\mathbf{HB}^{=}$-derivations without open assumption classes and with respectively the same conclusions in a bottom-up manner: we state the main theorem first, and prove it only later after the statement of a main lemma; this lemma will in its turn be proven only subsequently after the formulation of a further needed lemma; and so on in a similar way. The reason for this interleaving bottom-up approach in presenting our proof and in developing the transformation here consists in the amount of necessary technical details, which could quite easily deter the attention from our goal of establishing the existence of the desired transformation.

The proof of the main theorem of this section, Theorem 7.2.1 just below, uses a number of technical lemmas, of which the proof of Lemma 7.2.3 will actually contain the effective method of finding mimicking derivations in $\mathbf{HB}^{=}$ for the applications of the rules $\text{CONTRACT}^{-}_{(\text{nd})}$ and $\text{UFP}^{-}_{(\text{nd})}$. The proof of Lemma 7.2.9 employs, in a somewhat implicit way, the method from the completeness-proof of Brandt and Henglein. Lemma 7.2.6 follows from Lemma 7.2.9 by an additional lemma, Lemma 7.2.13, concerning the possibility to generate new $\mathbf{HB}^{=}_{\mathbf{0}}$-derivations from given ones by performing certain kinds of substitutions throughout a $\mathbf{HB}^{=}_{\mathbf{0}}$-derivation. Lemma 7.2.6 is the main tool for proving Lemma 7.2.3.

Theorem 7.2.1 (Effective transformation of $\mathbf{AC}^{=}$- and $\mathbf{AC}^{=}_{*}$-derivations into $\mathbf{HB}^{=}$-derivations). *Every derivation $\mathcal{D}$ in one of the systems $\mathbf{AC}^{=}$ or $\mathbf{AC}^{=}_{*}$ without assumptions can be transformed effectively into a derivation $\mathcal{D}'$ in $\mathbf{HB}^{=}$ with the same conclusion and without open assumption classes.*

We will *prove* this theorem subsequently on page 230. But first we introduce natural-deduction-system versions $\text{CONTRACT}^{-}_{(\text{nd})}$ and $\text{UFP}^{-}_{(\text{nd})}$ of the rules CONTRACT and UFP, give a proposition related to these rules, and formulate two lemmas that are concerned with what is the central issue to be settled in this section for the proof of Theorem 7.2.1: the elimination of applications of $\text{CONTRACT}^{-}_{(\text{nd})}$ and $\text{UFP}^{-}_{(\text{nd})}$ from derivations in the extension of $\mathbf{HB}^{=}$ with these rules.

We start by introducing the following natural-deduction-system variants of the rules UFP and CONTRACT from the systems $\mathbf{AC}^{=}$ and $\mathbf{AC}^{=}_{*}$. In a natural-deduction system $\mathcal{S}$ that has $\mu Tp\text{-}Eq$ as its set of formulas, the rule $\text{UFP}^{-}_{(\text{nd})}$ enables all applications that, at the bottom of a derivation in $\mathcal{S}+\text{UFP}^{-}_{(\text{nd})}$, are of the form

$$\frac{\begin{matrix}\mathcal{D}_1 \\ \tau_1 = \tau[\tau_1/\alpha]\end{matrix} \qquad \begin{matrix}\mathcal{D}_2 \\ \tau_2 = \tau[\tau_2/\alpha]\end{matrix}}{\tau_1 = \tau_2}\;\text{UFP}^{-}_{(\text{nd})} \quad \text{(no side-conditions on } \mathcal{D}_1 \text{ and } \mathcal{D}_2\text{)} \tag{7.31}$$

where $\tau, \tau_1, \tau_2 \in \mu Tp$, $\alpha \in TVar$, and $\mathcal{D}_1, \mathcal{D}_2 \in \mathcal{D}er(\mathcal{S})$; this rule does not allow other applications. And similarly, the rule $\text{CONTRACT}^{-}_{(\text{nd})}$ enables all applications that, at the bottom of a derivation in $\mathcal{S}+\mathrm{UFP}^{-}_{(\mathrm{nd})}$, are of the form

$$\frac{\begin{matrix}\mathcal{D}_1\\ \tau_1 = \tau[\tau_1/\alpha]\end{matrix}}{\tau_1 = \mu\alpha.\,\tau}\;\mathrm{CONTRACT}^{-}_{(\mathrm{nd})} \quad (\text{no side-condition on } \mathcal{D}_1) \tag{7.32}$$

where $\tau, \tau_1 \in \mu Tp$, $\alpha \in TVar$, and $\mathcal{D}_1 \in \mathcal{D}er(\mathcal{S})$; $\mathrm{CONTRACT}^{-}_{(\mathrm{nd})}$ does not allow other applications. The symbol $^{-}$ in the rule names $\mathrm{UFP}^{-}_{(\mathrm{nd})}$ and $\mathrm{CONTRACT}^{-}_{(\mathrm{nd})}$ is intended to indicate, similarly as in the case of the rule $\mathrm{DECOMP}^{-}_{(\mathrm{nd})}$ introduced in Chapter 5, Section 5.3, that applications of these rules are *not* subject to any side-conditions on the occurring open assumptions in immediate subderivations. Since the statements we are going to show apply even to these unrestricted natural-deduction-system versions of UFP and CONTRACT, we do not need to introduce here more restrictive natural-deduction-system versions $\mathrm{UFP}_{(\mathrm{nd})}$ and $\mathrm{CONTRACT}_{(\mathrm{nd})}$ (analogously to the version $\mathrm{DECOMP}_{(\mathrm{nd})}$ of DECOMP that was introduced in Chapter 5, Section 5.3), applications of which would be subject to the side-condition that no open assumptions are present.

The proposition below asserts that every derivation $\mathcal{D}$ without assumptions in $\mathbf{AC}^{=}$, or in $\mathbf{AC}^{=}_{*}$, can be transformed in an immediate way into a derivation $\tilde{\mathcal{D}}$ in $\mathbf{HB}^{=}+\mathrm{UFP}^{-}_{(\mathrm{nd})}$, or respectively in $\mathbf{HB}^{=}+\mathrm{CONTRACT}^{-}_{(\mathrm{nd})}$, such that $\mathcal{D}'$ mimics $\mathcal{D}$. This statement will be used in the proof of Theorem 7.2.1 to justify the second step of our transformation from $\mathbf{AC}^{=}$- and $\mathbf{AC}^{=}_{*}$-derivations into $\mathbf{HB}^{=}$-derivations (in the case of $\mathbf{AC}^{=}$-derivations, confer the illustration in Figure 7.1).

Proposition 7.2.2. *(i) Let $\mathcal{D}$ be a derivation in $\mathbf{AC}^{=}_{\_}$ without assumptions. The result $\tilde{\mathcal{D}}$ of renaming each application of* UFP *in $\mathcal{D}$ into an application of* $\mathrm{UFP}^{-}_{(\mathrm{nd})}$*, i.e. the prooftree that results from $\mathcal{D}$ by changing the rule name label of applications of* UFP *into* $\mathrm{UFP}^{-}_{(\mathrm{nd})}$*, is a derivation in $\mathbf{HB}^{=}+\mathrm{UFP}^{-}_{(\mathrm{nd})}$ without open assumptions and with the same conclusion as $\mathcal{D}$.*

(ii) Similarly, every derivation $\mathcal{D}$ in $\mathbf{AC}^{=}_{*\_}$ without assumptions can be transformed, by merely changing rule labels at applications of CONTRACT *in $\mathcal{D}$ from* CONTRACT *to* $\mathrm{CONTRACT}^{-}_{(\mathrm{nd})}$*, into a derivation $\tilde{\mathcal{D}}$ in $\mathbf{HB}^{=}+$ $+\mathrm{CONTRACT}^{-}_{(\mathrm{nd})}$ without open assumptions and with the same conclusion as $\mathcal{D}$.*

Proof. Both statements of the proposition are immediate consequences of the fact that all axioms of $\mathbf{AC}^{=}_{\_}$ and of $\mathbf{AC}^{=}_{*\_}$ are also axioms of $\mathbf{HB}^{=}$, and that all rules of $\mathbf{AC}^{=}_{\_}$ and of $\mathbf{AC}^{=}_{*\_}$ except the rules UFP and CONTRACT are also rules of $\mathbf{HB}^{=}$. There is, however, a conceptual difference between, for example, the rule SYMM in $\mathbf{AC}^{=}_{\_}$ or in $\mathbf{AC}^{=}_{*\_}$, and the rule SYMM in $\mathbf{HB}^{=}$: the first one is a pure-Hilbert-system rule, whereas the second one is a natural-deduction-system rule.[3] But both

[3] In the notation we use here frequently for such natural-deduction-system variants of pure-Hilbert-system rules, applications of which do not take into account the presence or absence

statements of the lemma can be shown in a straightforward way by induction on the depth $|\mathcal{D}|$ of derivations $\mathcal{D}$ without open assumptions in $\mathbf{AC}^{=}_{-}$, or in $\mathbf{AC}^{=}_{*-}$. □

The first of the mentioned lemmas is concerned with the elimination of such applications of $\text{CONTRACT}^{-}_{(\text{nd})}$and $\text{UFP}^{-}_{(\text{nd})}$ that occur at the bottom of derivations in $\mathbf{HB}^{=} + \text{CONTRACT}^{-}_{(\text{nd})} + \text{UFP}^{-}_{(\text{nd})}$. And the subsequent second lemma formulates a statement about the elimination of all applications of these two rules from arbitrary given derivations in $\mathbf{HB}^{=} + \text{CONTRACT}^{-}_{(\text{nd})} + \text{UFP}^{-}_{(\text{nd})}$.

Lemma 7.2.3. *The rules* $\text{CONTRACT}^{-}_{(\text{nd})}$ *and* $\text{UFP}^{-}_{(\text{nd})}$ *are derivable rules of* $\mathbf{HB}^{=}$. *In particular, applications of these two rules can effectively be eliminated from the bottom of derivations in* $\mathbf{HB}^{=} + \text{CONTRACT}^{-}_{(\text{nd})} + \text{UFP}^{-}_{(\text{nd})}$, *more precisely, the following two statements hold:*

(i) Every derivation $\mathcal{D}$ *in* $\mathbf{HB}^{=} + \text{CONTRACT}^{-}_{(\text{nd})} + \text{UFP}^{-}_{(\text{nd})}$, *which possibly contains open assumptions, that is of the form*

$$\dfrac{\begin{array}{c}\mathcal{D}_1\\ \tau_1 = \tau[\tau_1/\alpha]\end{array}}{\tau_1 = \mu\alpha.\,\tau}\ \text{CONTRACT}^{-}_{(\text{nd})}\,, \tag{7.33}$$

where $\mathcal{D}_1$ *is a derivation in* $\mathbf{HB}^{=}$, *can effectively be transformed into a derivation* $\mathcal{D}'$ *in* $\mathbf{HB}^{=}$ *of the form*

$$\begin{array}{c}\mathcal{D}_1\\ [\,\tau_1 = \tau[\tau_1/\alpha]\,]_1\\ \mathcal{DC}^{(\iota)}_{mim}\\ \tau_1 = \mu\alpha.\,\tau\end{array} \tag{7.34}$$

where the derivation context $\mathcal{DC}^{(\iota)}_{mim} \in \mathcal{D}er\mathcal{C}txt_1(\mathbf{HB}^{=})$ *does not contain open marked assumptions (i.e.* $\mathsf{massm}(\mathcal{DC}^{(\iota)}_{mim}) = \emptyset$ *holds), such that* $\mathcal{D}'$, *the result of filling context-holes* $[\,]_1$ *in* $\mathcal{DC}^{(\iota)}_{mim}$ *with the derivation* $\mathcal{D}_1$, *has the same conclusion and the same open assumption classes as* $\mathcal{D}$. *In particular, it holds that the derivation context* $\mathcal{DC}^{(\iota)}_{mim}$ *can effectively be produced from the application* ι *of* $\text{CONTRACT}^{-}_{(\text{nd})}$ *at the bottom of* $\mathcal{D}$.

(ii) Every derivation in $\mathbf{HB}^{=} + \text{UFP}^{-}_{(\text{nd})} + \text{CONTRACT}^{-}_{(\text{nd})}$, *with possibly open assumptions, that is of the form*

$$\dfrac{\begin{array}{c}\mathcal{D}_1\\ \tau_1 = \tau[\tau_1/\alpha]\end{array}\qquad\begin{array}{c}\mathcal{D}_2\\ \tau_2 = \tau[\tau_2/\alpha]\end{array}}{\tau_1 = \tau_2}\ \text{UFP}^{-}_{(\text{nd})}\,, \tag{7.35}$$

of open assumptions in immediate subderivations, the rule SYMM in $\mathbf{HB}^{=}$ could actually be denoted by $\text{SYMM}^{-}_{(\text{nd})}$ referring to the rule SYMM of $\mathbf{AC}^{=}$; also in this case the symbol $^{-}$ in the rule designation $\text{SYMM}^{-}_{(\text{nd})}$ would be intended to express (and make it abundantly clear) that applications of this rule in $\mathbf{HB}^{=}$ are not subject to any side-conditions on the open assumptions occurring in immediate subderivations.

where $\mathcal{D}_1$ and $\mathcal{D}_2$ are derivations in $\mathbf{HB}^=$, can effectively be transformed into a derivation $\mathcal{D}'$ in $\mathbf{HB}^=$ of the form

$$\begin{array}{c} \begin{array}{cc} \mathcal{D}_1 & \mathcal{D}_2 \\ [\,\tau_1 = \tau[\tau_1/\alpha]\,]_1 \qquad & [\,\tau_2 = \tau[\tau_2/\alpha]\,]_2 \end{array} \\ \mathcal{DC}^{(\iota)}_{mim} \\ \tau_1 = \tau_2 \end{array} \tag{7.36}$$

where $\mathcal{DC}^{(\iota)}_{mim} \in \mathcal{D}er\mathcal{C}txt_2(\mathbf{HB}^=)$ with $\mathsf{massm}(\mathcal{DC}^{(\iota)}_{mim}) = \emptyset$, such that $\mathcal{D}'$, the result of the filling context-holes $[\,]_1$ and $[\,]_2$ in $\mathcal{DC}^{(\iota)}_{mim}$ respectively with the derivations $\mathcal{D}_1$ and $\mathcal{D}_2$, has the same conclusion and the same open assumption classes as $\mathcal{D}$. Again, the derivation context $\mathcal{DC}^{(\iota)}_{mim}$ can effectively be produced from the application ι of $\mathrm{UFP}^-_{(\mathrm{nd})}$ *at the bottom of $\mathcal{D}$.*

We will *prove* this lemma below on page 233. But first we proceed with stating and proving the second lemma mentioned above, and then with giving a proof of Theorem 7.2.1 that relies on this lemma for justifying the third step of the transformation developed here from $\mathbf{AC}^=$- and $\mathbf{AC}^=_*$-derivations into $\mathbf{HB}^=$-derivations.

Lemma 7.2.4. *Applications of* $\mathrm{CONTRACT}^-_{(\mathrm{nd})}$ *and of* $\mathrm{UFP}^-_{(\mathrm{nd})}$ *can effectively be eliminated from derivations in the extension of $\mathbf{HB}^=$ with these rules. More precisely, every derivation $\mathcal{D}$ in $\mathbf{HB}^= + \mathrm{CONTRACT}^-_{(\mathrm{nd})} + \mathrm{UFP}^-_{(\mathrm{nd})}$ can be transformed effectively, by successively eliminating arbitrarily chosen applications of the rules* $\mathrm{CONTRACT}^-_{(\mathrm{nd})}$ *or* $\mathrm{UFP}^-_{(\mathrm{nd})}$*, into a derivation $\mathcal{D}'$ in $\mathbf{HB}^=$ with the same conclusion and with the same open assumption classes as $\mathcal{D}$.*

Sketch of Proof. The statement of the lemma can be shown by straightforward induction on the number of occurrences of applications of $\mathrm{CONTRACT}^-_{(\mathrm{nd})}$ and $\mathrm{UFP}^-_{(\mathrm{nd})}$ in a derivation in $\mathbf{HB}^= + \mathrm{CONTRACT}^-_{(\mathrm{nd})} + \mathrm{UFP}^-_{(\mathrm{nd})}$, where in the induction step the transformations guaranteed by Lemma 7.2.3 are applied to an arbitrarily picked application of $\mathrm{CONTRACT}^-_{(\mathrm{nd})}$ or $\mathrm{UFP}^-_{(\mathrm{nd})}$ in such a derivation. □

Proof of Thm. 7.2.1. Let $\mathcal{S}$ be one of the systems $\mathbf{AC}^=$ or $\mathbf{AC}^=_*$, and let $\mathcal{D}$ be an arbitrary derivation in $\mathcal{S}$ without assumptions and with conclusion $\tau = \sigma$, for some $\tau, \sigma \in \mu Tp$. We denote by $\mathcal{S}_-$ the system $\mathbf{AC}^=_-$, if $\mathcal{S} = \mathbf{AC}^=$, and the system $\mathbf{AC}^=_{*-}$, if $\mathcal{S} = \mathbf{AC}^=_-$.

The derivation $\mathcal{D}$ can effectively be transformed into a derivation $\mathcal{D}'$ in $\mathbf{HB}^=$ with the same conclusion and without open assumptions by performing the following three steps (we refer here again to the illustrations we have already given, namely, to Figure 7.1, for the case $\mathcal{S} = \mathbf{AC}^=$, and to Figure 7.2, for the case $\mathcal{S} = \mathbf{AC}^=_*$):

(1) μ-COMPAT-*elimination step:* Using the effective transformation stated by Corollary 7.1.17, transform the derivation $\mathcal{D}$ in $\mathcal{S}$ (by stepwise and effective elimination of the applications of μ-COMPAT in $\mathcal{D}$) into a derivation $\mathcal{D}^{(1)}$ in the system $\mathcal{S}_-$ without assumptions and with the same conclusion as $\mathcal{D}$.

For recognizing what task remains to be carried out in further steps, we ignore for a moment the conceptual difference between rules of the same name in the Hilbert-style proof systems $\mathcal{S}$ and $\mathcal{S}_{-}$, on the one hand, and in the natural-deduction-style system $\mathbf{HB}^{=}$, on the other hand, and in doing so we notice the following: The systems $\mathcal{S}_{-}$ and $\mathbf{HB}^{=}$ have the same axioms and differ, with respect to their inference rules, only in the rules UFP or CONTRACT (which are rules of $\mathcal{S}_{-}$, but not of $\mathbf{HB}^{=}$) and ARROW/FIX (which is a rule of $\mathbf{HB}^{=}$, but not of $\mathcal{S}_{-}$). This suggest the following possibility for constructing a derivation in $\mathbf{HB}^{=}$ without open assumptions and with conclusion $\tau = \sigma$ from $\mathcal{D}^{(1)}$: view $\mathcal{D}^{(1)}$ as a derivation $\mathcal{D}^{(2)}$ without open assumptions in a corresponding natural-deduction system, and then eliminate from $\mathcal{D}^{(2)}$ all those applications of natural-deduction system rules that correspond to the rules UFP and CONTRACT.

(2) *Renaming step of applications of* UFP *and* CONTRACT *into applications of* UFP$^{-}_{(\mathrm{nd})}$ *and* CONTRACT$^{-}_{(\mathrm{nd})}$*:* Replace all applications of the rules UFP or of CONTRACT in $\mathcal{D}^{(1)}$ by applications of the natural-deduction-system variants UFP$^{-}_{(\mathrm{nd})}$ and CONTRACT$^{-}_{(\mathrm{nd})}$ of UFP and CONTRACT, respectively. Due to Proposition 7.2.2, the result is a derivation $\mathcal{D}^{(2)}$ in the natural-deduction system $\mathbf{HB}^{=} + \mathrm{UFP}^{-}_{(\mathrm{nd})} + \mathrm{CONTRACT}^{-}_{(\mathrm{nd})}$ such that $\mathcal{D}^{(2)}$ has the same conclusion as $\mathcal{D}$ and such that it does not contain open assumptions.

It remains to eliminate all applications of UFP$^{-}_{(\mathrm{nd})}$ and CONTRACT$^{-}_{(\mathrm{nd})}$ from $\mathcal{D}^{(2)}$ in a further step.

(3) UFP$^{-}_{(\mathrm{nd})}$/CONTRACT$^{-}_{(\mathrm{nd})}$*-elimination step:* By applying the effective transformation stated by Lemma 7.2.4, eliminate successively (relying on the elimination steps described by by Lemma 7.2.3) all applications of UFP$^{-}_{(\mathrm{nd})}$ or of CONTRACT$^{-}_{(\mathrm{nd})}$ from $\mathcal{D}^{(2)}$. The result is then a derivation $\mathcal{D}'$ in $\mathbf{HB}^{=}$ without open assumptions and with the same conclusion $\tau = \sigma$ as $\mathcal{D}$.

In this effective way we have found a derivation $\mathcal{D}'$ in $\mathbf{HB}^{=}$ that contains no open assumptions and that has the same conclusion as the derivation $\mathcal{D}$ in $\mathcal{S}$, which we have assumed to be arbitrary.

□

Remark 7.2.5. By taking a closer look at the proof above, it is easy to notice that the second and third steps of our transformation (cf. the schematic illustration in Figure 7.1 and in Figure 7.2) can also be carried out for derivations containing assumptions. More precisely, it can be shown analogously that every derivation $\mathcal{D}^{(1)}$ in $\mathbf{AC}^{=}_{-}$ or in $\mathbf{AC}^{=}_{*-}$ can effectively be transformed into a derivation $\mathcal{D}'$ in $\mathbf{HB}^{=}$ such that the inhabited open assumption classes of $\mathcal{D}'$ correspond bijectively to occurrences of assumptions in $\mathcal{D}$, where this correspondence relates, for all $\rho, \chi \in \mu Tp$ and assumption markers u, an inhabited open assumption class $[\rho = \chi]^{u}$ of $\mathcal{D}'$ with a particular occurrence of the assumption $\rho = \chi$ in $\mathcal{D}$.

The reason for this is that Lemma 7.2.3 asserts that the unrestricted natural-deduction-system versions $\text{UFP}^{-}_{(\text{nd})}$ and $\text{CONTRACT}^{-}_{(\text{nd})}$ of the rules UFP and CONTRACT are derivable in $\mathbf{HB}^{=}$, and not just that the restricted natural-deduction-system versions $\text{UFP}_{(\text{nd})}$ and $\text{CONTRACT}_{(\text{nd})}$ (which can only be applied in the absence of open assumptions) of UFP and CONTRACT are derivable in $\mathbf{HB}^{=}$. An equivalent reason is that the rules $\text{UFP}^{-}_{(\text{nd})}$ and $\text{CONTRACT}^{-}_{(\text{nd})}$ are not just admissible, but even derivable in $\mathbf{HB}^{=}$, as stated by Lemma 7.2.3; this makes it possible to directly translate an arbitrary derivation $\mathcal{D}^{(2)}$ in $\mathbf{HB}^{=}+$ $+\,\text{CONTRACT}^{-}_{(\text{nd})}+\text{UFP}^{-}_{(\text{nd})}$ into a mimicking derivations $\mathcal{D}'$ in $\mathbf{HB}^{=}$ also in the case that $\mathcal{D}^{(2)}$ contains open assumptions.

For showing Theorem 7.2.1, however, a weaker version of Lemma 7.2.3 would actually have been sufficient that only stated derivability in $\mathbf{HB}^{=}$ of the rules $\text{UFP}_{(\text{nd})}$ and $\text{CONTRACT}_{(\text{nd})}$, or equivalently, that only stated admissibility of the rules $\text{UFP}^{-}_{(\text{nd})}$ and $\text{CONTRACT}^{-}_{(\text{nd})}$ in $\mathbf{HB}^{=}$.

The proof we give for Lemma 7.2.3 below, which will contain an effective method to "translate" applications of rules $\text{UFP}^{-}_{(\text{nd})}$ and $\text{CONTRACT}^{-}_{(\text{nd})}$ into the system $\mathbf{HB}^{=}$, makes essential use of the following technical Lemma 7.2.6 and will be given subsequently to the formulation of this auxiliary statement.

Lemma 7.2.6. *Let $\tau \in \mu Tp$ and $\alpha \in TVar$ such that $\alpha \downarrow \tau$ and $\alpha \in fv(\tau)$ holds. Furthermore, let $\sigma_1, \sigma_2 \in \mu Tp$ such that σ_1 and σ_2 are substitutible for α in τ.*

Then there exists a derivation $\mathcal{D}_{\tau,\alpha;\,\sigma_1,\sigma_2}$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ of the form

$$\begin{array}{cl}
[\sigma_1 = \sigma_2]^u & \\
\mathcal{D}_{\tau,\alpha;\,\sigma_1,\sigma_2} & \\
\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots & \text{ARROW or ARROW/FIX} \\
\tau^{(0)}[\sigma_1/\alpha] = \tau^{(0)}[\sigma_2/\alpha] & \\
::::::::::::::::::::::::::: & (\text{FOLD}_l,\ \text{FOLD}_r,\ \text{REN})^* \\
\tau[\sigma_1/\alpha] = \tau[\sigma_2/\alpha] &
\end{array} \tag{7.37}$$

with some $\tau^{(0)} \in \mu Tp$, some assumption marker u, and where $[\sigma_1 = \sigma_2]^u$ is the single open assumption class in this derivation that is inhabited. Hereby $\mathcal{D}_{\tau,\alpha;\,\sigma_1,\sigma_2}$ denotes the entire displayed derivation with conclusion $\tau[\sigma_1/\alpha] = \tau[\sigma_2/\alpha]$; dotted lines have been used to indicate rule applications at the bottom of this derivation to prevent the impression that $\mathcal{D}_{\tau,\alpha;\,\sigma_1,\sigma_2}$ ends at the displayed application of ARROW *or of* ARROW/FIX. *The indicated application of* ARROW *or* ARROW/FIX *is always present in $\mathcal{D}_{\tau,\alpha;\,\sigma_1,\sigma_2}$, whereas the number of following, possibly multiple, applications of* FOLD_l, FOLD_r *and* REN *might be zero.*

Moreover, given τ, σ_1, σ_2 and α as assumed above, a $\mathbf{HB}_{\mathbf{0}}^{=}$-derivation of the form (7.37) *can effectively be constructed.*

The *proof* for this lemma is given towards the end of this section on page 253. This proof will rely on another important technical lemma, Lemma 7.2.9 below, as well as on a further statement, Lemma 7.2.13, which is concerned with the 'asymmetric' substitution of two recursive types on the left- and right-hand sides of all equations between recursive types in a $\mathbf{HB}^{=}_{\mathbf{0}}$-derivation. Before we continue with the proof of Lemma 7.2.3, we formulate, as an aside, a rather obvious consequence of Lemma 7.2.6.

Proposition 7.2.7. *For all $\tau \in \mu Tp$ and for all $\alpha \in TVar$, the substitution rule* $\mathrm{SUBST}_{\tau[./\alpha]}$ *is derivable in* $\mathbf{HB}^{=}_{\mathbf{0}}$.

Proof. Let $\tau \in \mu Tp$ and $\alpha \in TVar$ be arbitrary. For showing that the substitution rule $\mathrm{SUBST}_{\tau[./\alpha]}$ is derivable in $\mathbf{HB}^{=}_{\mathbf{0}}$, we consider an arbitrary application of this rule of the form (7.13) with some $\sigma_1, \sigma_2 \in \mu Tp$.

If $\alpha \notin \mathrm{fv}(\tau)$, then this application can be mimicked by an axiom $\tau = \tau$ of $\mathbf{HB}^{=}_{\mathbf{0}}$.

If $\alpha \not\downarrow \tau$ holds, then $\tau \equiv \mu\alpha_1 \ldots \alpha_n . \alpha$ for some $n \in \omega$ and $\alpha_1, \ldots, \alpha_n \in TVar$ with $\alpha_1, \ldots, \alpha_n \not\equiv \alpha$. Since, due to the admissibility of the substitution expressions $\tau[\sigma_1/\alpha]$ and $\tau[\sigma_2/\alpha]$ in the conclusion of the considered $\mathrm{SUBST}_{\tau[./\alpha]}$-application, $\alpha_1, \ldots, \alpha_n$ do not occur free in σ_1 and σ_2, the considered application can be mimicked by a sequence of n FOLD_l- and n FOLD_r-applications with conclusion $\mu\alpha_1 \ldots \alpha_n . \sigma_1 = \mu\alpha_1 \ldots \alpha_n . \sigma_2$ and with single assumption $\sigma_1 = \sigma_2$.

If, however, $\alpha \in \mathrm{fv}(\tau)$ and $\alpha \downarrow \tau$ holds, then the considered application of $\mathrm{SUBST}_{\tau[./\alpha]}$ can be mimicked by the derivation $\mathcal{D}_{\tau,\alpha;\sigma_1,\sigma_2}$ in $\mathbf{HB}^{=}_{\mathbf{0}}$ that is guaranteed to exist by Lemma 7.2.6.

□

Proof of Lemma 7.2.3. Derivability of the rules $\mathrm{CONTRACT}^{-}_{(\mathrm{nd})}$ and $\mathrm{UFP}^{-}_{(\mathrm{nd})}$ in $\mathbf{HB}^{=}$ follow from stronger formulations (i)′ and (ii)′ of the statements (i) and (ii) in the lemma. More precisely, derivability of $\mathrm{CONTRACT}^{-}_{(\mathrm{nd})}$ in $\mathbf{HB}^{=}$ means, according to Definition 4.3.2, (iv), the statement

(i)′ Every derivation $\mathcal{D}$ in an extension by enlargement $\mathcal{S}_{\mathrm{ext}}$ such that $\mathcal{D}$ is of the form (7.33) with $\mathcal{D}_1 \in \mathcal{D}er(\mathcal{S}_{\mathrm{ext}})$ can effectively be transformed into a derivation $\mathcal{D}'$ in $\mathcal{S}_{\mathrm{ext}}$ of the form (7.34) where $\mathcal{DC}^{(\iota)}_{\mathrm{mim}} \in \mathcal{D}er\mathcal{C}txt_1(\mathbf{HB}^{=})$ such that $\mathcal{D}'$ has the same conclusion and the same open assumption classes as $\mathcal{D}$.

Similarly, a statement (ii)′ that expresses that $\mathrm{UFP}^{-}_{(\mathrm{nd})}$ is derivable in $\mathbf{HB}^{=}$ can be formulated as a stronger version of item (ii) of the lemma. The assertions (i)′ and (ii)′ can be shown (entirely) analogously to items (i) and (ii) in the lemma. Therefore we only show here the assertions (i) and (ii) of the lemma; this is done in items (a) and (b) below, respectively.

(a) Let $\mathcal{D}$ be an arbitrary derivation in $\mathbf{HB}^{=} + \mathrm{CONTRACT}^{-}_{(\mathrm{nd})} + \mathrm{UFP}^{-}_{(\mathrm{nd})}$ of the form (7.33), for some $\tau, \tau_1 \in \mu Tp$ and $\alpha \in TVar$, and for some derivation $\mathcal{D}_1$ in $\mathbf{HB}^{=}$. Then $\alpha \downarrow \tau$ holds as the side-condition on the application of $\mathrm{CONTRACT}^{-}_{(\mathrm{nd})}$ at the bottom of $\mathcal{D}$. We have to show that a derivation context $\mathcal{DC}^{(\iota)}_{\mathrm{mim}} \in \mathcal{D}er\mathcal{C}txt_1(\mathbf{HB}^{=})$ with $\mathsf{massm}(\mathcal{DC}^{(\iota)}_{\mathrm{mim}}) = \emptyset$ can effectively be found such that the derivation (7.34), which is built by filling the context-hole $[\,]_1$ in $\mathcal{DC}^{(\iota)}_{\mathrm{mim}}$ by $\mathcal{D}_1$, has the same conclusion and the same open assumption classes as $\mathcal{D}$.

Case 1. $\alpha \notin \mathrm{fv}(\tau)$.

In this case the application ι of $\mathrm{CONTRACT}^{-}_{(\mathrm{nd})}$ at the bottom of $\mathcal{D}$ corresponds to an application of the $\mathbf{HB}^{=}_{\mathbf{0}}$-rule FOLD_r. An application of this rule can be mimicked in $\mathbf{HB}^{=}$ by an application of TRANS with an axiom (FOLD/UNFOLD) and an application of SYMM just above its right premise. In particular, the desired derivation context $\mathcal{DC}^{(\iota)}_{\mathrm{mim}} \in \mathcal{D}er\mathcal{C}txt_1(\mathbf{HB}^{=})$ can here be chosen as

$$\dfrac{[\,\tau_1 = \tau\,]_1 \qquad \dfrac{\dfrac{\text{(FOLD/UNFOLD)}}{\mu\alpha.\,\tau = \underbrace{\tau[\mu\alpha.\,\tau/\alpha]}_{\Leftrightarrow\,\tau}}}{\tau = \mu\alpha.\,\tau}\,\text{SYMM}}{\tau_1 = \mu\alpha.\,\tau}\,\text{TRANS}$$

(where for reading convenience the premise $\tau_1 = \tau$ of ι has been indicated inside the context-hole $[]_1$ in $\mathcal{DC}^{(\iota)}_{\mathrm{mim}}$); obviously $\mathsf{massm}(\mathcal{DC}^{(\iota)}_{\mathrm{mim}}) = \emptyset$ holds for $\mathcal{DC}^{(\iota)}_{\mathrm{mim}}$. With this derivation context, the derivation of the form (7.34) has the same conclusion and the same open assumption classes as $\mathcal{D}$.

Case 2. $\alpha \in \mathrm{fv}(\tau)$.

For the proof of the existence, in this case, of the desired derivation context $\mathcal{DC}^{(\iota)}_{\mathrm{mim}}$, we will make the simplifying assumption that $\mu\alpha.\,\tau$ is substitutible for α in τ. If this is not the case, then our argumentation below can be used, as can be verified easily, to find an appropriate derivation context $\mathcal{DC}^{(\iota)}_{\mathrm{mim}}$ also in this situation, employing the observation that $\mathcal{D}$ can first be transformed into a mimicking derivation for $\mathcal{D}$ of the form

$$\dfrac{\dfrac{\dfrac{\begin{array}{c}\mathcal{D}_1\\ \tau_1 = \tau[\tau_1/\alpha]\end{array}}{\tau_1 = \tau'[\tau_1/\alpha]}\,\text{REN}}{\tau_1 = \mu\alpha.\,\tau'}\,\text{CONTRACT}^{-}_{(\text{nd})}}{\tau_1 = \mu\alpha.\,\tau}\,\text{REN}\ ,$$

where $\tau' \in \mu Tp$ is such that $\tau' \equiv_{\mathrm{ren}} \tau$ and that both τ_1 and $\mu\alpha.\,\tau$ (as well as then also $\mu\alpha.\,\tau'$) are substitutible for α in τ'.

We therefore assume now that $\mu\alpha.\,\tau$ is substitutible for α in τ. Now we first build a derivation context $\tilde{\mathcal{DC}}^{(\iota)}_{\mathrm{mim}} \in \mathcal{D}er\mathcal{C}txt_1(\mathbf{HB}^{=}_{\mathbf{0}}+\mathrm{TRANS})$ of the form shown in Figure 7.3: the part-derivation $\mathcal{D}'_{\tau,\alpha;\,\tau_1,\mu\alpha.\,\tau}$ occurring there is the result of renaming appropriately, if this is necessary, the assumption markers in discharged assumptions in the derivation $\mathcal{D}_{\tau,\alpha;\,\tau_1,\mu\alpha.\,\tau}$, which can effectively be found due to Lemma 7.2.6 (this is applicable due to $\alpha \in \mathrm{fv}(\tau)$ and $\alpha \downarrow \tau$), such that there is no assumption marker used in discharged assumptions of $\mathcal{D}'_{\tau,\alpha;\,\tau_1,\mu\alpha.\,\tau}$ that occurs as a marker for open assumptions in $\mathcal{D}_1$. And furthermore, the assumption marker $\boldsymbol{v}$ used in $\tilde{\mathcal{DC}}^{(\iota)}_{\mathrm{mim}}$ does not occur in $\mathcal{D}_1$ nor in $\mathcal{D}'_{\tau,\alpha;\,\tau_1,\mu\alpha.\,\tau}$.

Figure 7.3: Mimicking derivation context $\tilde{\mathcal{DC}}^{(\iota)}_{\mathrm{mim}}$ in $\mathbf{HB}^{=}_{\mathbf{0}}+\mathrm{TRANS}$ for an application ι of $\mathrm{CONTRACT}^{-}_{(\mathrm{nd})}$ at the bottom of a derivation $\mathcal{D}$ in $\mathbf{HB}^{=}+$ $+\,\mathrm{CONTRACT}^{-}_{(\mathrm{nd})}$ of the form (7.33), given that $\alpha\in\mathrm{fv}(\tau)$ holds for this application. For reading convenience, the premise $\tau_1=\tau[\tau_1/\alpha]$ of ι is indicated inside of occurrences of the hole $[\,]_1$ in $\tilde{\mathcal{DC}}^{(\iota)}_{\mathrm{mim}}$.

$$
\dfrac{\big[\,\tau_1=\tau[\tau_1/\alpha]\,\big]_1 \qquad
\dfrac{\begin{array}{c}
\dfrac{\big[\,\tau_1=\tau[\tau_1/\alpha]\,\big]_1 \qquad \dfrac{\dfrac{\big(\,\tau^{(0)}[\tau_1/\alpha]=\tau^{(0)}[\mu\alpha.\,\tau/\alpha]\,\big)^{\boldsymbol{v}}}{\tau[\tau_1/\alpha]=\tau[\mu\alpha.\,\tau/\alpha]}\,(\mathrm{FOLD}_{l/r},\,\mathrm{REN})^*}{\tau[\tau_1/\alpha]=\mu\alpha.\,\tau}\,\mathrm{FOLD}_r}{[\tau_1=\mu\alpha.\,\tau]}\,\mathrm{TRANS}\\
\mathcal{D}'_{\tau,\alpha;\,\tau_1,\mu\alpha.\,\tau}\\
\cdots\cdots\cdots\cdots\cdots\cdots\ \mathrm{ARROW/FIX},\,\boldsymbol{v}\\
\tau^{(0)}[\tau_1/\alpha]=\tau^{(0)}[\mu\alpha.\,\tau/\alpha]\\
\vdots\ (\mathrm{FOLD}_{l/r},\,\mathrm{REN})^*\\
\tau[\tau_1/\alpha]=\tau[\mu\alpha.\,\tau/\alpha]
\end{array}}{\tau[\tau_1/\alpha]=\mu\alpha.\,\tau}\,\mathrm{FOLD}_r}{\tau_1=\mu\alpha.\,\tau}\,\mathrm{TRANS}
$$

This derivation context $\tilde{\mathcal{DC}}^{(\iota)}_{\mathrm{mim}}$ obviously fulfills $\mathsf{massm}(\tilde{\mathcal{DC}}^{(\iota)}_{\mathrm{mim}})=\emptyset$ since $\mathcal{D}_{\tau,\alpha;\,\tau_1,\mu\alpha.\,\tau}$, and hence also $\mathcal{D}'_{\tau,\alpha;\,\tau_1,\mu\alpha.\,\tau}$, contains only the open assumption class $[\tau_1=\mu\alpha.\,\tau]^u$ for some marker u, and also because the occurrence of the marked assumption $(\,\tau^{(0)}[\tau_1/\alpha]=\tau^{(0)}[\mu\alpha.\,\tau/\alpha]\,)^{\boldsymbol{v}}$ at the top of $\tilde{\mathcal{DC}}^{(\iota)}_{\mathrm{mim}}$ is discharged within $\tilde{\mathcal{DC}}^{(\iota)}_{\mathrm{mim}}$. Furthermore, $\tilde{\mathcal{DC}}^{(\iota)}_{\mathrm{mim}}$ contains at least two occurrences of the context-hole $[\,]_1$; in fact it follows, since the assumption class[4] $[\tau_1=\mu\alpha.\,\tau]^u$ at the top of $\mathcal{D}_{\tau,\alpha;\,\tau_1,\mu\alpha.\,\tau}$, and hence also in $\mathcal{D}'_{\tau,\alpha;\,\tau_1,\mu\alpha.\,\tau}$, is inhabited, that there are actually $1+n$ many context-holes $[\,]_1$ in $\tilde{\mathcal{DC}}^{(\iota)}_{\mathrm{mim}}$, where m is the number of assumptions in the open assumption class $[\tau_1=\mu\alpha.\,\tau]^u$ of $\mathcal{D}'_{\tau,\alpha;\,\tau_1,\mu\alpha.\,\tau}$.

By utilizing the transformation described in the proof of Lemma 5.1.19, the derivation context $\tilde{\mathcal{DC}}^{(\iota)}_{\mathrm{mim}}\in\mathcal{D}er\mathcal{C}txt_1(\mathbf{HB}^{=}_{\mathbf{0}})$ can effectively be transformed into a corresponding derivation context $\mathcal{DC}^{(\iota)}_{\mathrm{mim}}\in\mathcal{D}er\mathcal{C}txt_1(\mathbf{HB}^{=})$ with the property $\mathsf{massm}(\mathcal{DC}^{(\iota)}_{\mathrm{mim}})=\emptyset$, with the same conclusion, and with at least two occurring context-holes $[\,]_1$. With this derivation context $\mathcal{DC}^{(\iota)}_{\mathrm{mim}}$ it holds that the derivation of the form (7.34) is a derivation in $\mathbf{HB}^{=}_{\mathbf{0}}$ with the same conclusion and with the same open assumption classes as $\mathcal{D}$.

[4] Within $\tilde{\mathcal{DC}}^{(\iota)}_{\mathrm{mim}}$ the marker u used in this assumption class disappears because there the part-derivation $\mathcal{D}'_{\tau,\alpha;\,\tau_1,\mu\alpha.\,\tau}$ of $\tilde{\mathcal{DC}}^{(\iota)}_{\mathrm{mim}}$ gets enlarged above all of the open assumptions $(\tau_1=\mu\alpha.\,\tau)^u$.

(b) Item (ii) of the lemma can now be shown immediately by using item (i) as follows.

We assume a derivation $\mathcal{D}$ in $\mathbf{HB}^{=}_{\mathbf{0}} + \mathrm{UFP}^{-}_{(\mathrm{nd})} + \mathrm{CONTRACT}^{-}_{(\mathrm{nd})}$ of the form (7.35), which ends with an application of $\mathrm{UFP}^{-}_{(\mathrm{nd})}$ and where $\mathcal{D}_1$ and $\mathcal{D}_2$ are two derivations in $\mathbf{HB}^{=}$, to be given. We have to show that it is possible to find effectively a derivation context $\mathcal{DC}^{(\iota)}_{\mathrm{mim}} \in \mathcal{D}er\mathcal{C}txt_2(\mathbf{HB}^{=})$ with $\mathsf{massm}(\mathcal{DC}^{(\iota)}_{\mathrm{mim}}) = \emptyset$ such that the derivation (7.36), which is built by filling the context-holes $[\,]_1$ and $[\,]_2$ in $\mathcal{DC}^{(\iota)}_{\mathrm{mim}}$ respectively by $\mathcal{D}_1$ and by $\mathcal{D}_2$, has the same conclusion and the same open assumption classes as $\mathcal{D}$.

Similarly as UFP-applications can be mimicked by derivations in the extension $\mathbf{AC}^{=}$+CONTRACT of $\mathbf{AC}^{=}$ (cf. Proposition 5.1.6), arbitrary applications of $\mathrm{UFP}^{-}_{(\mathrm{nd})}$ can be mimicked by derivations in $\mathbf{HB}^{=} + \mathrm{CONTRACT}^{-}_{(\mathrm{nd})}$, the extension of $\mathbf{HB}^{=}$ with the rule $\mathrm{CONTRACT}^{-}_{(\mathrm{nd})}$. Hence the derivation $\mathcal{D}$ can obviously be transformed into a derivation $\tilde{\mathcal{D}}$ of the form

$$\dfrac{\dfrac{\begin{array}{c}\mathcal{D}_1\\ \tau_1 = \tau[\tau_1/\alpha]\end{array}}{\tau_1 = \mu\alpha.\,\tau}\,\mathrm{CONTRACT}^{-}_{(\mathrm{nd})} \qquad \dfrac{\dfrac{\begin{array}{c}\mathcal{D}_2\\ \tau_2 = \tau[\tau_2/\alpha]\end{array}}{\tau_2 = \mu\alpha.\,\tau}\,\mathrm{CONTRACT}^{-}_{(\mathrm{nd})}}{\mu\alpha.\,\tau = \tau_2}\,\mathrm{SYMM}}{\tau_1 = \tau_2}\,\mathrm{TRANS}$$

with the same conclusion and with the same open assumption classes as $\mathcal{D}$. Now we let $\mathcal{DC}^{(\iota_1)}_{\mathrm{mim}}$ and $\mathcal{DC}^{(\iota_2)}_{\mathrm{mim}}$ be derivation-contexts in $\mathcal{D}er\mathcal{C}txt_1(\mathbf{HB}^{=})$ that can effectively be found due to item (i) of the lemma for the two applications ι_1 and ι_2 of $\mathrm{CONTRACT}^{-}_{(\mathrm{nd})}$in $\tilde{\mathcal{D}}$, which immediately follow the subderivations $\mathcal{D}_1$ and $\mathcal{D}_2$, such that $\mathcal{DC}^{(\iota_1)}_{\mathrm{mim}}$ and $\mathcal{DC}^{(\iota_2)}_{\mathrm{mim}}$ that fulfill $\mathsf{massm}(\mathcal{DC}^{(\iota_1)}_{\mathrm{mim}}) = \emptyset$ and $\mathsf{massm}(\mathcal{DC}^{(\iota_2)}_{\mathrm{mim}}) = \emptyset$, and such that, by the construction in item (a) of this proof, they respectively contain at least one occurrence of the context-hole $[\,]_1$. These two derivation contexts can now be arranged to form the new derivation context $\mathcal{DC}^{(\iota)}_{\mathrm{mim}} \in \mathcal{D}er\mathcal{C}txt_2(\mathbf{HB}^{=})$ of the form

$$\dfrac{\mathcal{DC}^{(\iota_1)}_{\mathrm{mim}} \qquad \dfrac{\mathcal{DC}^{(\iota_2)}_{\mathrm{mim}}[\,[\,]_2\,/\,[\,]_1\,]}{\mu\alpha.\,\tau = \tau_2}\,\mathrm{SYMM}}{\tau_1 = \tau_2}\,\mathrm{TRANS}$$

where $\mathcal{DC}^{(\iota_2)}_{\mathrm{mim}}[\,[\,]_2\,/\,[\,]_1\,]$ is the result of replacing all context-holes $[]_1$ in $\mathcal{DC}^{(\iota_2)}_{\mathrm{mim}}$ by the context-hole $[\,]_2$. It follows that $\mathsf{massm}(\mathcal{DC}^{(\iota)}_{\mathrm{mim}}) = \emptyset$ and that $\mathcal{DC}^{(\iota)}_{\mathrm{mim}}$ contains at least one occurrence each of the context-holes $[\,]_1$ and $[\,]_2$.

Due to this, the derivation context $\mathcal{DC}^{(\iota)}_{\mathrm{mim}}$ is of the desired form: it follows that the derivation of the form (7.36) is a derivation in $\mathbf{HB}^{=}$ with the same

Figure 7.4: Mimicking derivation context $\tilde{\mathcal{DC}}^{(\iota)}_{\text{mim}}$ in $\mathbf{HB}^=_{\mathbf{0}}$ + TRANS for an application ι of $\text{UFP}^-_{(\text{nd})}$ at the bottom of a derivation $\mathcal{D}$ in $\mathbf{HB}^=$ + $\text{UFP}^-_{(\text{nd})}$ of the form (7.35), given that $\alpha \in \text{fv}(\tau)$ holds for this application. For reading convenience, the premises $\tau_1 = \tau[\tau_1/\alpha]$ and $\tau_2 = \tau[\tau_2/\alpha]$ of ι have been indicated inside of occurrences in $\tilde{\mathcal{DC}}^{(\iota)}_{\text{mim}}$ of the context-holes $[\,]_1$ and $[\,]_2$, respectively.

$$
\dfrac{[\,\tau_1 = \tau[\tau_1/\alpha]\,]_1 \qquad \dfrac{\begin{array}{c}
\dfrac{[\,\tau_1 = \tau[\tau_1/\alpha]\,]_1 \qquad \dfrac{\begin{array}{c}\big(\,\tau^{(0)}[\tau_1/\alpha] = \tau^{(0)}[\tau_2/\alpha]\,\big)^{\boldsymbol{v}} \\ \vdots\ (\text{FOLD}_{l/r},\ \text{REN})^* \\ \tau[\tau_1/\alpha] = \tau[\tau_2/\alpha]\end{array} \qquad \dfrac{[\,\tau_2 = \tau[\tau_2/\alpha]\,]_2}{\tau[\tau_2/\alpha] = \tau_2}\,\text{SYMM}}{\tau[\tau_1/\alpha] = \tau_2}\,\text{TRANS}}{[\tau_1 = \tau_2]}\,\text{TRANS} \\
\mathcal{D}'_{\tau,\alpha;\,\tau_1,\tau_2} \\
\vdots\ \text{ARROW/FIX},\ \boldsymbol{v} \\
\tau^{(0)}[\tau_1/\alpha] = \tau^{(0)}[\tau_2/\alpha] \\
\vdots\ (\text{FOLD}_{l/r},\ \text{REN})^* \\
\tau[\tau_1/\alpha] = \tau[\tau_2/\alpha]
\end{array} \qquad \dfrac{[\,\tau_2 = \tau[\tau_2/\alpha]\,]_2}{\tau[\tau_2/\alpha] = \tau_2}\,\text{SYMM}}{\tau[\tau_1/\alpha] = \tau_2}\,\text{TRANS}}{\tau_1 = \tau_2}\,\text{TRANS}
$$

conclusion and the same open assumption classes as $\mathcal{D}$. And our construction clearly also shows that such derivation contexts $\mathcal{DC}^{(\iota)}_{\text{mim}}$ can be found by applying an effective procedure to a given application ι of $\text{UFP}^-_{(\text{nd})}$ at the bottom of a derivation $\mathcal{D}$ in $\mathbf{HB}^=$ + $\text{UFP}^-_{(\text{nd})}$ of the form (7.35).

□

Remark 7.2.8. We want to mention that the detour made in the proof of item (ii) of the lemma via the proof of (i) is not inherently necessary. This is because a bottommost application of $\text{UFP}^-_{(\text{nd})}$ in a derivation $\mathcal{D}$ of the form (7.35) can also be eliminated effectively in a very similar way as done so for the elimination of an application of $\text{CONTRACT}^-_{(\text{nd})}$ in the proof of item (i) of the lemma. In particular, for applications ι of $\text{UFP}^-_{(\text{nd})}$ at the bottom of a derivation $\mathcal{D}$ of the form (7.35) with the property that $\alpha \in \text{fv}(\tau)$ holds, a mimicking derivation context $\mathcal{DC}^{(\iota)}_{\text{mim}}$ in $\mathcal{DerCtxt}_2(\mathbf{HB}^=_{\mathbf{0}})$ can be found from $\tilde{\mathcal{DC}}^{(\iota)}_{\text{mim}} \in \mathcal{DerCtxt}_2(\mathbf{HB}^=_{\mathbf{0}} + \text{SYMM} + \text{TRANS})$, the derivation context depicted in Figure 7.4. There, the part-derivation $\mathcal{D}'_{\tau,\alpha;\tau_1,\tau_2}$ that appears in $\tilde{\mathcal{DC}}^{(\iota)}_{\text{mim}}$ arises from a derivation $\mathcal{D}_{\tau,\alpha;\tau_1,\tau_2}$ in $\mathbf{HB}^=_{\mathbf{0}}$, which is guaranteed by Lemma 7.2.6, by appropriate renamings of the assumption markers of the discharged assumption classes in $\mathcal{D}_{\tau,\alpha;\tau_1,\tau_2}$ such that no marker of an open assumption class in $\mathcal{D}_1$ or $\mathcal{D}_2$ coincides with an assumption marker for a discharged assumption class in $\mathcal{D}'_{\tau,\alpha;\tau_1,\tau_2}$. And the assumption marker $\boldsymbol{v}$ is different from all assumption markers in either of $\mathcal{D}_1$, $\mathcal{D}_2$ or $\mathcal{D}'_{\tau,\alpha;\tau_1,\tau_2}$.

The next lemma will be our most important tool in the proof of Lemma 7.2.6. Its own proof, which we are going to give in rather much detail, employs methods from the completeness-proof of Brandt and Henglein for their coinductively motivated axiomatizations of the relations $=_\mu$ and $\leq_\mu$.

Lemma 7.2.9. *For all $\tau \in \mu Tp$ and $\alpha \in TVar$ such that $\alpha \downarrow \tau$ and $\alpha \in fv(\tau)$ holds, and for all $\sigma_1, \sigma_2 \in \mu Tp$ that are substitutible for α in τ, there exists a derivation $\mathcal{D}^{(\sigma_1,\sigma_2)}_{\tau,\alpha}$ in $\mathbf{HB}^{=}_{\mathbf{0}}$ of the form*

$$\left.\begin{array}{ll} \left[\dfrac{(\mathrm{REFL})}{\alpha = \alpha}\right] & \\ \mathcal{D}^{(\sigma_1,\sigma_2)}_{\tau,\alpha} & \\ \cdots\cdots\cdots\cdots & \text{ARROW or ARROW/FIX} \\ \tau^{(0)} = \tau^{(0)} & \\ ::::::::::::: & (\mathrm{FOLD}_l,\ \mathrm{FOLD}_r,\ \mathrm{REN})^*\,, \\ \tau = \tau & \end{array}\right\} \tag{7.38}$$

where

(i) $\mathcal{D}^{(\sigma_1,\sigma_2)}_{\tau,\alpha}$ does not contain open assumptions,

(ii) for all formulas $\chi_1 = \chi_2$ in $\mathcal{D}^{(\sigma_1,\sigma_2)}_{\tau,\alpha}$, the recursive types σ_1 and σ_2 are substitutible for α in both χ_1 and χ_2,

(iii) there is at least one occurrence of an axiom (REFL) *of the form $\alpha = \alpha$ at the top of $\mathcal{D}^{(\sigma_1,\sigma_2)}_{\tau,\alpha}$ (i.e. the class of axioms $\left[\dfrac{(\mathrm{REFL})}{\alpha = \alpha}\right]$ indicated at the top of the symbolic prooftree* (7.38) *for $\mathcal{D}^{(\sigma_1,\sigma_2)}_{\tau,\alpha}$ is inhabited),*

(iv) $\mathcal{D}^{(\sigma_1,\sigma_2)}_{\tau,\alpha}$ does not contain, in leaves at its top, any axioms (REFL) *of the form $\chi = \chi$ such that $\chi \not\equiv \alpha$ and $\alpha \in fv(\chi)$ holds,*

(v) as indicated through dotted lines at the bottom of the symbolic prooftree (7.38) *for $\mathcal{D}^{(\sigma_1,\sigma_2)}_{\tau,\alpha}$, this derivation contains at its bottom an application of* ARROW *or* ARROW/FIX *that is then followed immediately by zero, one, or more applications of* REN, FOLD_l *or* FOLD_r.

and which derivation can always can be found effectively for given τ, σ_1, σ_2 and α as above.

We *prove* this lemma, starting on page 239, immediately below the following two remarks. The first of these remarks explains a consequence of condition (iv) in Lemma 7.2.9 (given that condition (i) in the lemma is fulfilled). And the second remark sketches a similarity between the proof of Lemma 7.2.9 that we will give below and the completeness proof given by Brandt and Henglein in [BrHe98].

Remark 7.2.10. Under the assumptions on τ, σ_1, σ_2 and α that are stipulated in Lemma 7.2.9, it is a consequence of the conditions (i) and (iv) on the derivation $\mathcal{D}^{(\sigma_1,\sigma_2)}_{\tau,\alpha}$, which is stated to exist there, that for all occurrences of equations between recursive types $\chi_1 = \chi_2$, where $\chi_1, \chi_2 \in \mu Tp$, in leaves at the top of the prooftree

$\mathcal{D}^{(\sigma_1,\sigma_2)}_{\tau,\alpha}$ the following holds: if α occurs free in χ_1 or in χ_2, then this formula occurrence takes place either within an axiom (REFL) of the form $\alpha = \alpha$ or within a marked assumption $(\chi_1 = \chi_2)^u$ that is discharged in $\mathcal{D}^{(\sigma_1,\sigma_2)}_{\tau,\alpha}$.

Remark 7.2.11. The proof of Lemma 7.2.9 that we will develop below is very similar to the proof given by Brandt and Henglein in [BrHe98] for the completeness of their axiomatization of the subtyping relation $\leq_\mu$ between recursive types; we will refer to this axiomatization here by $\mathbf{HB}^{\leq}$. In both proofs a desired derivation for a valid conclusion (i.e. an equation between recursive types $\tau = \tau$ such that for an $\alpha \in \mathit{TVar}$ $\alpha \in \mathrm{fv}(\tau)$ and $\alpha \downarrow \tau$ holds, in the case of the proof below of Lemma 7.2.9, and an inequality between recursive types $\tau \leq \sigma$ such that $\tau \leq_\mu \sigma$ holds, in the case of the proof in [BrHe98]) is built by a procedure that acts as following: It repeatedly extends a derivation, starting with the given conclusion and proceeding with intermediately reached derivations, in single steps to larger derivations, which usually are not yet of respectively desired form (of a derivation of the form (7.38) and as stated by Lemma 7.2.9, in our case, and of a derivation in $\mathbf{HB}^{\leq}$ without open assumptions, in the case treated in [BrHe98]).

These extension-steps consist in picking an appropriate leaf in the reached derivation, and in extending the prooftree above this leaf by a few more rule applications, or in binding back and discharging the formula at this leaf to a rule application occurring deeper down in the derivation. Although during the extension process as a whole there is a certain order imposed on which leaves at the top of intermediatel yreached derivations have to be chosen for immediate treatment, some limited cause for non-determinism is left in these choices; however, this has no effect on termination of the procedure.

In the case treated in [BrHe98] of the completeness proof for $\mathbf{HB}^{\leq}$ with respect to the relation $\leq_\mu$ it is possible to choose the extension-steps, starting from a given inequality $\tau \leq \sigma$ where $\tau, \sigma \in \mu Tp$ such that $\tau \leq_\mu \sigma$, in such a way that after finitely many of such steps a derivation in $\mathbf{HB}^{\leq}$ without open assumptions and with conclusion $\tau \leq \sigma$ is found effectively. And also in the analogous, but somewhat easier situation here of the proof below for Lemma 7.2.9, the extension-steps can be chosen in such a way that ultimately a derivation of the desired form (7.38) is always found after finitely many steps. The first of these extension steps is hereby applied to an axiom (REFL) $\tau = \tau$, where $\tau \in \mu Tp$ and $\alpha \in \mathit{TVar}$ are such that $\alpha \in \mathrm{fv}(\tau)$ and $\alpha \downarrow \tau$ holds. The produced derivation will here, in particular, be such that it contains no open assumptions and no axioms (REFL) of the form $\chi = \chi$ at the top with the properties $\chi \not\equiv \alpha$ and $\alpha \in \mathrm{fv}(\chi)$.

Notwithstanding these similarities between procedures in a proof in [BrHe98], and in the proof of Lemma 7.2.9 below, we will describe the respective extension-procedure here in a more proof-theoretically motivated way than this done so in [BrHe98].

Now we give the somewhat technical and lengthy proof for the Lemma 7.2.9.

Proof of Lemma 7.2.9. We assume recursive types $\tau, \sigma_1, \sigma_2 \in \mu Tp$ and $\alpha \in TVar$ to be given, such that

$$\alpha \in \mathrm{fv}(\tau),\ \alpha \downarrow \tau \text{ and } \sigma_1, \sigma_2 \text{ are substitutible for } \alpha \text{ in } \tau. \tag{7.39}$$

It is our purpose to produce a derivation $\mathcal{D}^{(\sigma_1,\sigma_2)}_{\tau,\alpha}$ of the form (7.38) with the properties (i)–(v) of the lemma.

We will first define a process P that is able to bring an intermediary derivation $\mathcal{D}$ in $\mathbf{HB}^=$ which is of a form in between an axiom (REFL) $\tau = \tau$ and the desired derivation $\mathcal{D}^{(\sigma_1,\sigma_2)}_{\tau,\alpha}$ in (7.38) by one simple step nearer to the desired form. P proceeds by choosing *such* a leaf at the top of an an intermediary derivation $\mathcal{D}$ *that* witnesses that $\mathcal{D}$ is not yet of the form (7.38) (for now, we will call such leaves to be "not of required form") and by then treating this in one of two possible ways: either by extending $\mathcal{D}$ above the considered leaf by some 1–3 more applications of $\mathbf{HB}^=_\mathbf{0}$-rules or by being able to discharge the equation between recursive types at that leaf in $\mathcal{D}$ (by binding it to an application of ARROW or ARROW/FIX deeper down in $\mathcal{D}$). P will be applicable to intermediary derivations as long as a derivation of the form (7.38) and as described in the lemma has not yet been found. Thereby it furthermore always chooses such a leaf with the property "not of the required form" that is *of minimal height* over the conclusion in the intermediary derivation.

If the process P is started on the initial derivation $\mathcal{D}^{(0)}$ consisting of just the axiom (REFL) $\tau = \tau$, and is then iterated on its outcomes, then a sequence $\langle \mathcal{D}^{(0)}, \mathcal{D}^{(1)}, \mathcal{D}^{(2)}, \ldots \rangle$ of derivations is produced that get larger and larger, and extend each other in the sense that they have common end-derivations[5]. It will turn out that if the process P^*, the iteration of P, is started on the derivation consisting of the axiom (REFL) $\tau = \tau$ then it will terminate in a desired derivation of the form (7.38) after finitely many executions of P. The most important argument for termination of P will be the following: the minimal heights in the derivations $\mathcal{D}^{(n)}$ of leaves that are not of the required form will increase steadily with the number of executions of P (though by far not necessarily with *every* execution of P) and would converge to infinity, if P^* did not terminate. Reasons for this are the precise definition of P in the following, which leads to the following features of this process: that (a) P always chooses *bottommost* leaves in a given (intermediary) derivation, which are not of required form, and that (b) for the finite or infinite sequence $\mathcal{D}^{(0)}, \mathcal{D}^{(1)}, \mathcal{D}^{(2)}, \ldots$ generated by the repeated application of P to the initial derivation $\tau = \tau$ it holds that, for all $i, j \in \omega$ with $j < i$, the derivation $\mathcal{D}^{(i)}$ is of larger size than the derivation $\mathcal{D}^{(j)}$, i.e. $\mathrm{s}(\mathcal{D}^{(i)}) > \mathrm{s}(\mathcal{D}^{(j)})$ holds.

Since the definition of the extension-steps performed by P will guarantee (together with a lemma on the finiteness of the number of subterms of a recursive type modulo taking variants), that there is in fact a bound depending on a quadratic function in the size $\mathrm{s}(\tau)$ of τ on the length of threads in the intermediary derivations $\mathcal{D}^{(n)}$ produced by P^* (this is described in Appendix A). This will finally show the

[5] This is a slight simplification here because it can happen that applications of ARROW in "common end-derivations" are changed into respective applications of ARROW, at which open assumptions are discharged, from a certain member of the sequence onwards once and forever.

termination of the application of P^* to the initial derivation consisting of just the axiom (REFL) $\tau = \tau$.

(1) We start by defining the process P:

The process P acts on all such $\mathbf{HB}_{\mathbf{0}}^{=}$-derivations $\mathcal{D}$, which with respect to the given $\tau, \sigma_1, \sigma_2 \in \mu Tp$ and $\alpha \in TVar$ with the property (7.39) fulfills the following properties (7.40) and (7.41) described below; on other derivations (or on other objects) the process $\mathcal{D}$ is undefined. The property (7.40) consists thereby of the following six assertions:

(a) $\mathcal{D}$ is a $\mathbf{HB}_{\mathbf{0}}^{=}$-derivation without open assumption classes and with the conclusion $\tau = \tau$.

(b) The leaves at the top of the prooftree $\mathcal{D}$ are either axioms (REFL) of $\mathbf{HB}_{\mathbf{0}}^{=}$ or marked assumptions that are discharged in $\mathcal{D}$.

(c) σ_1, σ_2 are substitutible for α in every equation between recursive types $\chi_1 = \chi_2$ occurring in $\mathcal{D}$.

(d) $\mathcal{D}$ fulfills the condition **AA** (cf. Definition C.3 in Appendix C).

(e) $\mathcal{D}$ does not contain nlμb-decreasing applications of FOLD$_{l/r}$ (see Definition C.9, Appendix C).

(f) $\mathcal{D}$ does not contain two successive applications of REN. Furthermore, at the top of $\mathcal{D}$ no application of REN takes place immediately below an occurrence of axiom (REFL) (however, applications of REN *are allowed* to follow *marked assumptions* in $\mathcal{D}$, also if these are of the (REFL)-axiom-like form $(\chi = \chi)^u$ in $\mathcal{D}$, where $\chi \in \mu Tp$ and u is an assumption marker).

As indicated above we will refer to the statement

$$\left.\text{"The above properties (a)–(f) are fulfilled with respect to } \mathcal{D},\ \alpha,\ \tau, \sigma_1 \text{ and } \sigma_2\text{."}\right\} \qquad (7.40)$$

by using its label (7.40). And furthermore the property (7.41) refers to the assertion:

$$\left.\text{"There is at least one occurrence of an axiom (REFL) of the form } \chi = \chi \text{ as leaf at the top of } \mathcal{D}, \text{ for which } \chi \not\equiv \alpha \text{ and } \alpha \in \mathrm{fv}(\chi) \text{ holds."}\right\} \qquad (7.41)$$

For a given derivation $\mathcal{D}$ with (7.40) and (7.41) the process P picks an arbitrary axiom $\chi = \chi$ in an unmarked formula at the top of $\mathcal{D}$ with the property, that $\chi \not\equiv \alpha$ and $\alpha \in \mathrm{fv}(\chi)$, and *which is moreover of minimal height* with this property in $\mathcal{D}$ (this entails that there sometimes are finitely many non-deterministic choices possible for the process P). By displaying, in boldface, the chosen axiom occurrence in a leaf, the derivation $\mathcal{D}$ can be written as

$$\begin{array}{c} \left(\dfrac{\textbf{(REFL)}}{\boldsymbol{\chi = \chi}}\right) \\ \mathcal{D} \\ \tau = \tau \end{array}$$

The process P will transform the derivation $\mathcal{D}$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ into a derivation $\mathcal{D}'$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ of either the form (7.42) or (7.43). In the first case $\mathcal{D}'$ is of the form

$$\begin{array}{c} \Big(\dfrac{\text{(REFL)}}{\chi_1 = \chi_1}\Big) \ \Big\{\Big(\dfrac{\text{(REFL)}}{\chi_2 = \chi_2}\Big)\Big\} \\ \mathcal{D}_e \\ (\boldsymbol{\chi} = \boldsymbol{\chi}) \\ \mathcal{D} \\ \tau = \tau \end{array} \tag{7.42}$$

where $\chi_1, \chi_2 \in \mu Tp$ are some generated subterms of χ which are reachable from χ by $\rightarrow_{\text{ren}}$-steps together with a single $\rightarrow_{\text{out-dec}}$- or $\rightarrow_{\text{out-unf}}$-reduction step, and where $\mathcal{D}_e$ is a short $\mathbf{HB}_{\mathbf{0}}^{=}$-derivation of between one and three rule applications with one or two new leaves arising; the possible second new axiom occurring at the top of $\mathcal{D}_e$ has been indicated by curly brackets. In the second case, $\mathcal{D}'$ is of the form

$$\begin{array}{c} \dfrac{(\chi' = \chi')^u}{(\boldsymbol{\chi} = \boldsymbol{\chi})}\,\text{REN} \\ \mathcal{D}^{(d)} \\ \tau = \tau \end{array} \tag{7.43}$$

where the marked assumption $(\chi' = \chi')^u$ is discharged at an application of (ARROW/FIX, u) in the derivation $\mathcal{D}^{(d)}$ that is the same as $\mathcal{D}$ except that possibly one application of ARROW in $\mathcal{D}$ has been changed to an application of (ARROW/FIX, u) at which in this case the marked hypothesis $(\chi' = \chi')^u$ at the top of $\mathcal{D}$ is then discharged.

In both cases the transformed derivation $\mathcal{D}'$ will again have the property (7.40); but in the second case the property (7.41) may have been lost during a *final* execution-step of P, in which (as it will turn out) a derivation of the desired form (7.38) and with the properties as described in the lemma will have been reached.

For the detailed description of these extensions we let $\mathcal{D}$ be a $\mathbf{HB}_{\mathbf{0}}^{=}$-derivation, which fulfills (7.40) and (7.41), and we fix an axiom occurrence $\chi = \chi$ in $\mathcal{D}$ that is of miminal height above the conclusion of $\mathcal{D}$ with the property that $\chi \not\equiv \alpha$ and $\alpha \in \text{fv}(\chi)$ hold.

For the definition of P, we distinguish the only two possible cases for χ that, on the one hand, χ is a composite recursive type $\chi_1 \rightarrow \chi_2$, for some $\chi_1, \chi_2 \in \mu Tp$, or that, on the other hand, χ starts with a μ-binding and is of the form $\mu\beta.\,\chi_0$, for some $\beta \in TVar$ and $\chi_0 \in \mu Tp$.

Case 1. $\chi \equiv \chi_1 \rightarrow \chi_2$ for some $\chi_1, \chi_2 \in \mu Tp$.

Subcase A: In the thread in $\mathcal{D}$ downwards from the considered axiom occurrence $\chi = \chi$ in a leaf there is an occurrence of the formula $\chi' = \chi'$ for some variant χ' of χ as the conclusion of an application of ARROW or ARROW/FIX.

For instance, with the considered axiom occurrence typeset in boldface, $\mathcal{D}$ is of the form

$$\left.\begin{array}{c}
\left(\dfrac{\textbf{(REFL)}}{\underbrace{\boldsymbol{\chi_1 \to \chi_2}}_{\equiv\chi} = \underbrace{\boldsymbol{\chi_1 \to \chi_2}}_{\equiv\chi}}\right)\\[2ex]
\begin{array}{ccc}\mathcal{D}_1 & \qquad\qquad\qquad & \mathcal{D}_2\end{array}\\
\dfrac{\chi'_1 = \chi'_1 \qquad\qquad\qquad \chi'_2 = \chi'_2}{(\underbrace{\chi'_1 \to \chi'_2}_{\equiv\chi'} = \underbrace{\chi'_1 \to \chi'_2}_{\equiv\chi'})}\ \text{ARROW}\\
\mathcal{DC}_0\\
\tau = \tau
\end{array}\right\}\qquad(7.44)$$

where $\chi' \equiv \chi'_1 \to \chi'_2$, $\chi'_1 \equiv_{\text{ren}} \chi_1$ and $\chi'_2 \equiv_{\text{ren}} \chi_2$ for some $\chi'_1, \chi'_2 \in$ $\in \mu Tp$, and where further the displayed application of ARROW is the *topmost* occurrence of an application of ARROW or ARROW/FIX in the thread down from this considered leaf that has a conclusion $\chi'' = \chi''$ for some $\chi'' \in \mu Tp$ with $\chi'' \equiv_{\text{ren}} \chi$.

In this situation the process P transforms $\mathcal{D}$ into the derivation $\mathcal{D}'$

$$\begin{array}{c}
\dfrac{(\chi'_1 \to \chi'_2 = \chi'_1 \to \chi'_2)^u}{\boldsymbol{(\chi_1 \to \chi_2 = \chi_1 \to \chi_2)}}\ \text{REN}\\
\begin{array}{ccc}\mathcal{D}_1 & \qquad\qquad\qquad & \mathcal{D}_2\end{array}\\
\dfrac{\chi'_1 = \chi'_1 \qquad\qquad\qquad \chi'_2 = \chi'_2}{(\chi'_1 \to \chi'_2 = \chi'_1 \to \chi'_2)}\ \text{ARROW/FIX}, u\\
\mathcal{DC}_0\\
\tau = \tau
\end{array}$$

(the position of the original leaf-occurrence of the axiom $\chi = \chi$ is still typeset in boldface) where u is a new[6] assumption marker, i.e. one that does not occur elsewhere in $\mathcal{D}$. For the sake of convenience in regard to a later argument, we use a (then trivial) application of REN for the extension here even in the case $\chi \equiv \chi'$, and hence also in a situation in which such an application would not be necessary to facilitate the back-binding.

If the considered axiom occurrence of $\chi = \chi$ in the subderivation $\mathcal{D}_2$ in a derivation of the form (7.44) (instead of in $\mathcal{D}_1$ as displayed there), then P acts by extending $\mathcal{D}$ above this leaf in an analogous way to a derivation $\mathcal{D}'$.

If $\mathcal{D}$ is of a similar form to (7.44), but with an application of ARROW/FIX, v instead of the displayed application of ARROW, then

[6] It would suffice to demand that u is not the marker of an assumption class (whether open or still undischarged) in $\mathcal{D}_1$ or in $\mathcal{D}_2$.

P acts similarly by discharging, in the transformed derivation $\mathcal{D}'$, a newly introduced marked assumption $(\chi_1' \to \chi_2' = \chi_1' \to \chi_2')^v$ at this application of ARROW/FIX, v. Generally, there is no need for a new assumption marker to be introduced in this case. *However*, if the assumption marker v is also used to discharge assumptions within $\mathcal{D}_1$ (i.e. if there occur assumptions in $\mathcal{D}_1$ marked by v that are discharged *within* $\mathcal{D}_1$), then we rename all open assumptions of the form $(\chi_1' \to \chi_2' = \chi_1' \to \chi_2')^v$ in $\mathcal{D}_1$ into open assumptions of the form $(\chi_1' \to \chi_2' = \chi_1' \to \chi_2')^w$ for some entirely fresh assumption marker w, and we also mark the newly arising leaf in $\mathcal{D}'$ by w.

If the considered axiom occurrence of $\chi = \chi$ takes place in the subderivation $\mathcal{D}_2$ in a derivation of the form (7.44) (instead of in $\mathcal{D}_1$ as displayed there), and the topmost application of ARROW or ARROW/FIX below this axiom occurrence is an application of ARROW/FIX, then P acts on $\mathcal{D}$ in an analogous way as described in the previous paragraph.

Clearly in all these situations the derivations $\mathcal{D}'$ produced by P again satisfy (7.40).

Subcase B: The condition in Subcase A does not hold for the derivation $\mathcal{D}$ and the considered axiom occurrence of $\chi = \chi$ at the top of $\mathcal{D}$.

Then P transforms $\mathcal{D}$ into the derivation $\mathcal{D}'$

$$\cfrac{\cfrac{\text{(REFL)}}{\chi_1 = \chi_1} \qquad \cfrac{\text{(REFL)}}{\chi_2 = \chi_2}}{(\underbrace{\boldsymbol{\chi_1 \to \chi_2}}_{\equiv \chi} = \underbrace{\boldsymbol{\chi_1 \to \chi_2}}_{\equiv \chi})}\ \text{ARROW}$$

$$\begin{array}{c}\mathcal{D}\\ \tau = \tau\end{array}$$

which again satisfies (7.40).

Case 2. $\chi \equiv \mu\beta.\, \chi_0$ for some $\chi_0 \in \mu Tp$.

Since $\alpha \in \mathrm{fv}(\chi)$, it must hold that $\beta \not\equiv \alpha$, and because furthermore σ_1, σ_2 are substitutible for α in $\mu\beta.\, \chi_0$, it follows that $\beta \notin \mathrm{fv}(\sigma_1) \cup \mathrm{fv}(\sigma_2)$. Due to $\alpha \in \mathrm{fv}(\chi_0)$ and $\alpha \not\equiv \beta$, it furthermore follows, due to Lemma 7.1.1, (i), that $\beta \downarrow \chi_0$.

Now let χ_0' be a variant of χ_0 such that (a) $\mu\beta.\, \chi_0$ (and as a consequence then also $\mu\beta.\, \chi_0'$) is substitutible for β in χ_0', and (b) σ_1 and σ_2 are substitutible for α and β in χ_0'. Then (since $\beta \notin \mathrm{fv}(\sigma_i)$) σ_i is substitutible for α in $\mu\beta.\, \chi_0'$ for each $i \in \{1, 2\}$. And furthermore, $(\mu\beta.\, \chi_0')[\sigma_i/\alpha]$ is (again due to $\beta \notin \mathrm{fv}(\sigma_i)$) substitutible for β in $\chi_0'[\sigma_i/\alpha]$ for each $i \in \{1, 2\}$. From $\beta \downarrow \chi_0$, which we saw above, it also follows that $\beta \downarrow \chi_0'$ by Lemma 7.1.2, (i).

Now due to $\alpha \not\equiv \beta$, $\beta \notin \mathrm{fv}(\sigma)$, and the admissibility of the substitution expressions $\chi_0'[\mu\beta.\, \chi_0'/\beta]$ and $(\mu\beta.\, \chi_0')[\sigma_i/\alpha]$ (which admissibility statements follow respectively from (a) and (b) above), an application of Lemma 3.3.13 gives that

$$\chi_0'[\mu\beta.\,\chi_0'/\beta]\,[\sigma_i/\alpha] \;\;\Leftrightarrow\;\; \chi_0'[\sigma_i/\alpha]\,[(\mu\beta.\,\chi_0')[\sigma_i/\alpha]/\beta]$$

holds for each $i \in \{1,2\}$. From above we know that the implicit side conditions on the right expression are fulfilled here. Hence we can conclude that σ_1 and σ_2 are then also substitutible for α in $\chi_0'[\mu\beta.\,\chi_0'/\beta]$.

Now the process P extends $\mathcal{D}$ above the considered leaf $\chi = \chi$ by three rule applications with the derivation $\mathcal{D}'$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ of the form

$$\begin{array}{c}
\dfrac{\dfrac{\dfrac{\dfrac{(\mathrm{REFL})}{\chi_0'[\mu\beta.\,\chi_0'/\beta] = \chi_0'[\mu\beta.\,\chi_0'/\beta]}\,\mathrm{FOLD}_r}{\chi_0'[\mu\beta.\,\chi_0'/\beta] = \mu\beta.\,\chi_0'}\,\mathrm{FOLD}_l}{\mu\beta.\,\chi_0' = \mu\beta.\,\chi_0'}\,\mathrm{REN}}{(\,\underbrace{\boldsymbol{\mu\beta.\,\chi_0}}_{\equiv\chi} = \underbrace{\boldsymbol{\mu\beta.\,\chi_0}}_{\equiv\chi}\,)} \\
\mathcal{D} \\
\tau = \tau
\end{array}$$

as the result (where the position in $\mathcal{D}'$ corresponding to the leaf in $\mathcal{D}$ above which $\mathcal{D}$ has been extended has again been typeset in boldface).

It is now easy to see, that $\mathcal{D}'$ again suffices condition (7.40). In particular it holds, due to $\beta \downarrow \chi_0$ which we have found above, that the two new applications of FOLD_l and of FOLD_r are not nlμb-decreasing.

Hereby we have concluded the definition of the process P.

Summarizing we notice the following property of the process P: for every derivation $\tilde{\mathcal{D}}$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ the assertion

$$\left.\begin{array}{l}
\tilde{\mathcal{D}} \text{ fulfills (7.40) and (7.41)} \;\Longrightarrow \\
\qquad\qquad \Longrightarrow \;\; \text{the result of applying } P \text{ to } \tilde{\mathcal{D}} \text{ fulfills (7.40)}
\end{array}\right\} \qquad (7.45)$$

holds.

(2) We will now show the following: if, for given $\tau, \sigma_1, \sigma_2 \in \mu Tp$ and $\alpha \in TVar$ such that (7.39) holds, the process P is started on the $\mathbf{HB}_{\mathbf{0}}^{=}$-derivation $\mathcal{D}^{(0)}$ consisting of the axiom $\tau = \tau$ (REFL), then it terminates after finitely many steps and it produces a derivation $\mathcal{D}$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ with the properties (7.40) and $\neg\,$(7.41) (the negation of (7.41)).

To prove this, we let $\tau, \sigma_1, \sigma_2 \in \mu Tp$ and $\alpha \in TVar$ be such that (7.39) holds, and we let $\mathcal{D}^{(0)}$ be the trivial $\mathbf{HB}_{\mathbf{0}}^{=}$-derivation consisting of just the axiom

$$\frac{(\mathrm{REFL})}{\tau = \tau} \qquad (7.46)$$

It is easy, indeed mostly trivial, to check that

$$\mathcal{D}^{(0)} \text{ satisfies (7.40) and (7.41).} \tag{7.47}$$

Now we define the sequence $\langle \mathcal{D}^{(n)} \rangle_{n\in\omega}$ of $\mathbf{HB}^=_\mathbf{0}$-derivations or of the symbol $\uparrow$ (with the meaning "undefined") that starts with $\mathcal{D}^{(0)}$ and is defined by the inductive clause

$$(\forall\, n \in \omega) \left[\mathcal{D}^{(n+1)} =_{\text{def}} \right.$$

$$\left. =_{\text{def}} \begin{cases} \text{result of} & \ldots\ \mathcal{D}^{(n)} \text{ is defined} \\ \quad\text{applying } P \text{ to } \mathcal{D}^{(n)} & \quad\ \text{and fulfills (7.40)} \\ \uparrow & \ldots\ \text{else} \end{cases} \right] . \tag{7.48}$$

In the following we consider only that finite or infinite subsequence $\mathcal{SD} =_{\text{def}} =_{\text{def}} \langle \mathcal{D}^{(n)} \rangle_{n\in I}$ of this sequence, which consists of $\mathcal{D}^{(0)}$ and all following $\mathbf{HB}^=_\mathbf{0}$-derivations $\mathcal{D}^{(n)}$ until for the first time $\uparrow$ is encountered. This means that I is an initial segment of ω containing zero, and it holds either that $I = \omega$, if $\mathcal{D}^{(n)}$ is defined by (7.48) for all $n \in \omega$, or that $I = \{0, 1, \ldots, n_{\max}\}$ for some $n_{\max} \in \omega$, if there exists a natural number j such that $\mathcal{D}^{(j)} = \uparrow$ and where $n_{\max}$ is the smallest natural number n such that $\mathcal{D}^{(n)}$ is defined, but $\mathcal{D}^{(n+1)}$ is undefined, i.e. $\mathcal{D}^{(n+1)} \equiv \uparrow$ holds.

We noticed after the definition of the process P in (1) that for every $\mathbf{HB}^=_\mathbf{0}$-derivation $\tilde{\mathcal{D}}$ the assertion (7.45) holds. From this and from (7.47) it follows, due to the inductive definition of the sequence $\langle \mathcal{D}^{(n)} \rangle_{n\in\omega}$ above (through (7.46) and (7.48) and the definition of I), by induction on n that it holds:

$$\left.\begin{array}{l} (\forall\, n \in I) \\ \quad \big[\big(n+1 \in I \implies \mathcal{D}^{(n)} \text{ satisfies (7.40) and (7.41)} \big) \ \& \\ \quad\ \ \& \ \big(n+1 \notin I \implies \mathcal{D}^{(n)} \text{ satisfies (7.40) and } \neg\,\text{(7.41)} \big) \,\big] \,. \end{array}\right\} \tag{7.49}$$

We further observe from the description of the process P in (1) that if P is applied to a derivation $\tilde{\mathcal{D}}$ such that (7.40) and (7.41) holds, then P extends $\tilde{\mathcal{D}}$ by between one and three additional rule applications to the result $\tilde{\mathcal{D}}'$; hence the size $\mathrm{s}(\tilde{\mathcal{D}}')$ of the prooftree $\tilde{\mathcal{D}}$ is always strictly greater than the size $\mathrm{s}(\tilde{\mathcal{D}})$ of $\tilde{\mathcal{D}}$. This implies for our sequence $\mathcal{SD}$ the following statement:

$$(\forall\, n \in I) \ \big[n+1 \in I \implies \mathrm{s}(\mathcal{D}^{(n+1)}) \geq \mathrm{s}(\mathcal{D}^{(n)}) + 1 \big] \tag{7.50}$$

We will now show that the sequence $\mathcal{SD}$ is actually finite.

Suppose that this is not the case, that $I = \omega$ and hence that $\mathcal{SD} = \langle \mathcal{D}^{(n)} \rangle_{n\in\omega}$. Then it follows from (7.50) that the sizes $\mathrm{s}(\mathcal{D}^{(n)})$ of the derivations $\mathcal{D}^{(n)}$ diverge against ∞, more symbolically, that

$$\langle \mathrm{s}(\mathcal{D}^{(n)}) \rangle_{n\in\omega} \to \infty \tag{7.51}$$

holds. Since the system $\mathbf{HB}^{=}_{\mathbf{0}}$ possesses only contain one- and two-premise rules (and that hence prooftrees of $\mathbf{HB}^{=}_{\mathbf{0}}$-derivations certainly are only finitely branching), this implies

$$\langle |\mathcal{D}^{(n)}| \rangle_{n\in\omega} \to \infty \tag{7.52}$$

as well, i.e. that also the depths $|\mathcal{D}^{(n)}|$ of the derivations $\mathcal{D}^{(n)}$ diverge against ∞. Due to $I = \omega$, it follows from (7.49) that (7.40) holds for all $\mathcal{D}^{(n)}$ with $n \in \omega$. Thus the assumptions of Theorem C.11 in Appendix C.11 are fulfilled for all $\mathcal{D}^{(n)}$. Hence this theorem can be applied here and it implies that

$$(\forall\, n \in \omega)\ \big[\, |\mathcal{D}^{(n)}| \ < \ 2\big((\mathrm{s}(\tau)+1)^2 + 8|\tau| + 2\,\big)\,\big] \tag{7.53}$$

holds, i.e. that there is a finite bound on the depth of the derivations $\mathcal{D}^{(n)}$ of the sequence $\mathcal{SD}$. This is an obvious contradiction to (7.52). Therefore our assumption that $\mathcal{SD}$ is infinite cannot be sustained.

In this way we have shown that the sequence $\mathcal{SD}$ is finite, i.e. that, for some $n_{\max} \in \omega$ and $I = \{0,\ldots,n_{\max}\}$, $\mathcal{SD} = \langle \mathcal{D}^{(n)} \rangle_{n\in I}$ is the case, and that the process P^* builds up, from the derivation $\mathcal{D}^{(0)}$ by precisely $n_{\max}$ iterations of P, a derivation $\mathcal{D}^{(n_{\max})}$ with the properties (7.40) and $\neg$(7.41) (this follows now from (7.49)).

(3) Finally we show: if the process P^* is started on the derivation $\mathcal{D}^{(0)}$ consisting only of the axiom (REFL) of the form $\tau = \tau$, then it terminates after finitely many steps and produces a derivation $\mathcal{D}^{(\sigma_1,\sigma_2)}_{\tau;\,\alpha}$ in $\mathbf{HB}^{=}_{\mathbf{0}}$ that is of the required form (7.38) with the properties (i)–(v) in the lemma.

We have shown in (2) that if the process P^* is started on the derivation $\mathcal{D}^{(0)}$ in $\mathbf{HB}^{=}_{\mathbf{0}}$ consisting only of the axiom (REFL) of the form $\tau = \tau$, then it produces a finite sequence $\langle \mathcal{D}^{(n)} \rangle_{n\in I}$ with $I = \{0,\ldots,n_{\max}\}$, for some $n_{\max} \in \omega\backslash\{0\}$, such that (7.49) holds and such that P^* terminates after exactly n executions of P due to the failure of condition (7.41) for $\mathcal{D}_{n_{\max}}$.

We let $\mathcal{D} =_{\mathrm{def}} \mathcal{D}^{(n_{\max})}$ and will demonstrate the assertion in this item by showing that $\mathcal{D}$ does actually fulfill the requirements (i)–(v) in the lemma, and hence that it can rightly be taken as the desired derivation $\mathcal{D}^{(\sigma_1,\sigma_2)}_{\tau,\,\alpha}$.

Since $\mathcal{D}$ satisfies (7.40), the requirements (i) and (ii) in the lemma are clearly fulfilled for $\mathcal{D}$. The requirement (iv) in the lemma holds, because it is the negation of (7.41), for which negation, $\neg$(7.41), we know that it holds for $\mathcal{D}$.

Hence it remains to show, that $\mathcal{D}$ is indeed of the form (7.38) (with $\mathcal{D}$ in place of $\mathcal{D}^{(\sigma_1,\,\sigma_2)}_{\tau,\,\alpha}$), where the displayed class of axioms $\alpha = \alpha$ at the top of the symbolic prooftree $\mathcal{D}$ gathers all occurrences of axioms of this form at the top of $\mathcal{D}$ and where this class is actually inhabited (non-empty). In particular, it suffices to show, that

$$\left.\begin{array}{l}\mathcal{D}\text{ contains one application of ARROW or of ARROW/FIX,}\\ \text{which towards the conclusion of }\mathcal{D}\text{ is only succeeded by a finite}\\ \text{number (that may be zero) of applications of rules FOLD}_{l/r}\\ \text{and/or rules REN,}\end{array}\right\} \tag{7.54}$$

and that

$$\left.\begin{array}{l}\text{there is at least one occurrence of an axiom (REFL) of the}\\ \text{form } \alpha = \alpha \text{ at the top of the prooftree } \mathcal{D}.\end{array}\right\} \tag{7.55}$$

First we show (7.54). We are going to consider the two cases, that $\mathrm{nl}\mu\mathrm{b}(\tau) = 0$ holds, and respectively, that $\mathrm{nl}\mu\mathrm{b}(\tau) > 0$ holds, separately below.

Case (i): $\mathrm{nl}\mu\mathrm{b}(\tau) = 0$.

Here τ must be of the form $\tau \equiv \tau_1 \to \tau_2$, for some $\tau_1, \tau_2 \in \mu Tp$, because due to the assumption (7.39) $\alpha \in \mathrm{fv}(\tau)$ and $\alpha \downarrow \tau$ holds. It follows that the derivation $\mathcal{D}^{(1)}$ built by the first application of P to $\mathcal{D}^1$ is actually of the form

$$\dfrac{\dfrac{(\text{REFL})}{\tau_1 = \tau_1} \qquad \dfrac{(\text{REFL})}{\tau_2 = \tau_2}}{\tau_1 \to \tau_2 = \tau_1 \to \tau_2}\ \text{ARROW} \tag{7.56}$$

since the process P applied to $\mathcal{D}^{(1)}$ has to act here according to Case 1, Subcase B, in its definition in (1). Since further executions of P on $\mathcal{D}^{(1)}$ extend this derivation only above its leaves, the premises and the conclusion of the last rule application in $\mathcal{D}$ are the same as, respectively, the premises and the conclusion of the last rule application in $\mathcal{D}^{(1)}$; however, it may be the case that the bottommost application of ARROW at the bottom of $\mathcal{D}^{(1)}$ is ultimately replaced by an application of ARROW/FIX, at which some assumptions of $\mathcal{D}$ get discharged. In any case $\mathcal{D}$ must then be of the form

$$\dfrac{\begin{array}{c}\mathcal{D}_1\\ \tau_1 = \tau_1\end{array} \qquad \begin{array}{c}\mathcal{D}_2\\ \tau_2 = \tau_2\end{array}}{\tau_1 \to \tau_2 = \tau_1 \to \tau_2}\ \text{ARROW or ARROW/FIX, } u \tag{7.57}$$

for some subderivations $\mathcal{D}_1$ and $\mathcal{D}_2$ (if the last rule application in $\mathcal{D}$ is one of ARROW/FIX with an assumption marker u attached to it, $\mathcal{D}_1$ and $\mathcal{D}_2$ must contain open assumptions of the form $[\tau_1 \to \tau_2 = \tau_1 \to \tau_2]^u$ that are discharged at the bottom of $\mathcal{D}$). Hence (7.54) holds for $\mathcal{D}$ in this case.

Case (ii): $\mathrm{nl}\mu\mathrm{b}(\tau) = m > 0$.

In this situation it follows that the first m derivations of the sequence $\langle \mathcal{D}^{(n)} \rangle_{n \in I}$, which is defined as sequence according to (7.46) and (7.48) and by removing all trailing symbols $\uparrow$ from the sequence if such occur, are all of the form

$$\begin{array}{cl}
(\text{REFL}) & \\ \hline
\tau_j = \tau_j & \\ \hline\hline
 & \text{FOLD}_{l/r},\ \text{REN} \\
\vdots & \\ \hline\hline
 & \text{FOLD}_{l/r},\ \text{REN} \\
\tau_1 = \tau_1 & \\ \hline\hline
 & \text{FOLD}_{l/r},\ \text{REN} \\
\tau = \tau &
\end{array}$$

for some $\tau_1, \dots, \tau_m \in \mu Tp$ with $\alpha \in \mathrm{fv}(\tau_j)$, $\alpha \downarrow \tau_j$ and $\mathrm{nl}\mu\mathrm{b}(\tau_j) = m - j$ for all $j \in \{1, \dots, m\}$. This is because the process P extends, in a $\mathbf{HB}^=_\mathbf{0}$-derivation, a leaf that carries an axiom (REFL) of the form $\mu\tilde{\alpha}.\,\tilde{\tau}_0 = \\ = \mu\tilde{\alpha}.\,\tilde{\tau}_0$ with the properties $\alpha \in \mathrm{fv}(\mu\tilde{\alpha}.\,\tilde{\tau}_0)$ and $\alpha \downarrow \mu\tilde{\alpha}.\,\tilde{\tau}_0$ according to Case 2 of the definition of P in (1) by a new subderivation that carries one new leaf of the form $\tilde{\tau}'[\mu\tilde{\alpha}.\,\tilde{\tau}'_0/\tilde{\alpha}] = \tilde{\tau}'[\mu\tilde{\alpha}.\,\tilde{\tau}'_0\,/\tilde{\alpha}]$ for some $\tilde{\tau}' \equiv_{\mathrm{ren}} \tilde{\tau}$ and with $\alpha \in \mathrm{fv}(\tilde{\tau}'[\mu\tilde{\alpha}.\,\tilde{\tau}'_0/\tilde{\alpha}])$, $\mathrm{nl}\mu\mathrm{b}(\tilde{\tau}'[\mu\tilde{\alpha}.\,\tilde{\tau}'_0/\tilde{\alpha}]) = \mathrm{nl}\mu\mathrm{b}(\mu\tilde{\alpha}.\,\tilde{\tau}'_0) - 1$ (cf. Lemma 3.5.7) and with $\alpha \downarrow \tilde{\tau}'[\mu\tilde{\alpha}.\,\tilde{\tau}'_0/\tilde{\alpha}]$ (cf. Lemma 7.1.2). – It follows that $\mathrm{nl}\mu\mathrm{b}(\tau_m) = 0$, $\alpha \in \mathrm{fv}(\tau_m)$ as well as $\alpha \downarrow \tau_m$, and hence that, for some $\tau_{m1}, \tau_{m2} \in \mu Tp$, $\tau_m \equiv \tau_{m1} \to \tau_{m2}$ is the case. And from this it is clear that the $(m+1)$-st extension-step of the process P^* by extending $\mathcal{D}^{(m)}$ to $\mathcal{D}^{(m+1)}$ is of the kind Case 1, Subcase B, in definition of P in (1); since furthermore P^* extends $\mathcal{D}^{(m)}$ only above its leaves this means that eventually $\mathcal{D}$ is of the form

$$\begin{array}{cl} \mathcal{D}_{a1} \qquad\qquad \mathcal{D}_{a2} & \\ \tau_{m1} = \tau_{m1} \qquad \tau_{m2} = \tau_{m2} & \\ \hline \tau_m = \tau_m & \text{ARROW or ARROW/FIX, } u \\ \hline\hline \tau = \tau & (\text{REN, FOLD}_{l/r})^{3m} \end{array} \tag{7.58}$$

with (possibly) some marker u and some subderivations $\mathcal{D}_{a1}$ and $\mathcal{D}_{a2}$ (with possibly the discharged assumption class $[\tau_m = \tau_m]^u$). Therefore (7.54) holds again in this case.

In this way we have shown the needed statement (7.54) about the form of $\mathcal{D}$ with an application of ARROW or ARROW/FIX at the bottom of $\mathcal{D}$, which is then only possibly followed by applications of one-premise rules of $\mathbf{HB}^=_\mathbf{0}$.

Now we are going to show (7.55), i.e. that there is indeed at least one axiom $\alpha = \alpha$ as an unmarked leaf at the top of $\mathcal{D}$.

To show this, we define the *minimal syntactical depth min-dp*$_{\tilde{\alpha}}(\tilde{\tau})$ *of a free occurrence of* the variable $\tilde{\alpha}$ *in* a recursive type $\tilde{\tau}$: for all $\tilde{\alpha} \in TVar$ and $\tilde{\tau} \in \mu Tp$, we define the positive integer $\text{min-dp}_{\tilde{\alpha}}(\tilde{\tau})$ by induction on the syntactical depth $|\tilde{\tau}|$ of $\tilde{\tau}$ using the clauses

$$\text{min-dp}_{\tilde{\alpha}}(\tilde{\tau}) =_{\mathrm{def}} \begin{cases} 0 & \dots\ |\tilde{\tau}| = 0 \ \text{ or } \ \tilde{\alpha} \notin \mathrm{fv}(\tilde{\tau})\,, \\ 1 + \min\bigl\{\text{min-dp}_{\tilde{\alpha}}(\tilde{\tau}_i) \mid i \in \{1,2\},\ \tilde{\alpha} \in \mathrm{fv}(\tilde{\tau}_i)\bigr\} & \\ & \dots\ \alpha \in \mathrm{fv}(\tilde{\tau}) \ \text{ and } \ \tilde{\tau} \equiv \tilde{\tau}_1 \to \tilde{\tau}_2\,, \\ 1 + \text{min-dp}_{\tilde{\alpha}}(\tilde{\tau}_0) & \dots\ \tilde{\alpha} \in \mathrm{fv}(\tilde{\tau}) \ \text{ and } \ \tilde{\tau} \equiv \mu\beta.\,\tilde{\tau}_0\,. \end{cases}$$

We will use the following three properties of this notion: for all $\tilde{\alpha}, \beta \in TVar$ and $\tilde{\tau}_1, \tilde{\tau}_2, \tilde{\tau}_0 \in \mu Tp$ it holds that

$$\tilde{\alpha} \in \mathrm{fv}(\tilde{\tau}_1 \to \tilde{\tau}_2) \implies (\exists\, i \in \{1,2\}) \bigl[\,\text{min-dp}_{\tilde{\alpha}}(\tilde{\tau}_1 \to \tilde{\tau}_2) = 1 + \text{min-dp}_{\tilde{\alpha}}(\tilde{\tau}_i)\,\bigr]\,, \tag{7.59}$$

$$\tilde{\tau}_1 \equiv_{\mathrm{ren}} \tilde{\tau}_2 \implies \text{min-dp}_{\tilde{\alpha}}(\tilde{\tau}_1) = \text{min-dp}_{\tilde{\alpha}}(\tilde{\tau}_2)\,, \tag{7.60}$$

$$\tilde{\alpha} \in \mathrm{fv}(\mu\beta.\,\tilde{\tau}_0) \implies \text{min-dp}_{\tilde{\alpha}}(\mu\beta.\,\tilde{\tau}_0) = 1 + \text{min-dp}_{\tilde{\alpha}}(\tilde{\tau}_0[\mu\beta.\,\tilde{\tau}_0/\beta])\,. \tag{7.61}$$

Hereby (7.59) follows immediately from the definition of min-dp$_{\tilde{\alpha}}(\cdot)$. (7.60) and (7.61) can be proven quite similar to analogous properties of the notion mμd$_{\tilde{\alpha}}(\cdot)$ of the minimal μ-depth of a variable in a recursive type $\tilde{\tau}$ used in the proof of Lemma 3.9.9, (i), in Appendix A, Section A.5, starting on page 341. In particular, (7.60) follows easily

$$\tilde{\alpha} \in \text{fv}(\tilde{\tau}) \;\;\&\;\; \beta \not\equiv \tilde{\alpha} \implies \text{min-dp}_{\tilde{\alpha}}((\tilde{\tau})) = \text{min-dp}_{\tilde{\alpha}}(\tilde{\tau}[\tilde{\sigma}/\beta])$$

Our main observation for the proof of (7.55) consists in the following assertion:

Suppose that $\tilde{\alpha} \in \mathit{TVar}$ and $\chi \in \mu Tp$ with $\tilde{\alpha} \in \text{fv}(\chi)$ and $\chi \not\equiv \tilde{\alpha}$ and suppose further that $\begin{array}{c}\mathcal{D}_e\\ \chi = \chi\end{array}$ is the extension-derivation above a picked leaf of the form $\chi = \chi$, which the process P produces in Case 1, Subcase B, or in Case 2 of its definition in (1).

$\implies$ There exists a leaf $\chi_0 = \chi_0$ in $\mathcal{D}_e$ with $\tilde{\alpha} \in \text{fv}(\chi_0)$ and with $\text{min-dp}_{\tilde{\alpha}}(\chi_0) = \text{min-dp}_{\tilde{\alpha}}(\chi) - 1$. (7.62)

This assertion follows very easily from checking the definition of P in Case 1, Subcase A and in Case 2 in item (1) above and by using the above properties (7.59), (7.60) and (7.61).

Since during the execution of the process P^* to $\mathcal{D}^{(0)}$ that is of the form of an axiom (REFL) $\tau = \tau$, where τ is such that (7.39) holds, in intermediary derivations $\mathcal{D}^{(n)}$ of the sequence $\langle \mathcal{D}^{(n)} \rangle_{n \in I}$ defined by (7.46) and (7.48) all occurring leaves $\chi = \chi$ with $\alpha \in \text{fv}(\chi)$ and $\alpha \not\equiv \chi$ are extended *at some stage* (in one of the three different ways described in (1)), the assertion (7.62) actually implies the following statement

There exists a thread Θ in $\mathcal{D}$ from the conclusion $\tau = \tau$ upwards on which, beginning from the conclusion $\chi_0 = \chi_0$ that is equal to the conclusion $\tau = \tau$ of $\mathcal{D}$, after respectively at most three rule applications the next following formula of the sequence

$$\xi =_{\text{def}} \langle\, (\chi_0 = \chi_0), (\chi_1 = \chi_1), \ldots, (\chi_{m-1} = \chi_{m-1}), (\chi_m = \chi_m) \,\rangle$$

can be found, where $m =_{\text{def}} \text{min-dp}_{\alpha}(\tau)$ and $\chi_0, \chi_1, \ldots, \chi_m \in$ $\in \mu Tp$, such that furthermore the following conditions are satisfied:

$$(\forall\, i \in \{0, \ldots, m\}) \left[\, \alpha \in \text{fv}(\chi_i) \,\right],$$
$$(\forall\, i \in \{0, \ldots, m-1\}) \left[\, \text{min-dp}_{\alpha}(\chi_{i+1}) = \text{min-dp}_{\alpha}(\chi_i) - 1 \,\right],$$
$$\text{min-dp}_{\alpha}(\chi_m) = 0\,. \qquad (7.63)$$

Hereby it is used that, in inductively building up the thread Θ with the sequence ξ of formulas on it in (7.63) via by successive respective extensions over leaves carrying equations $\chi = \chi$ with $\chi \not\equiv \alpha$ and $\alpha \in \text{fv}(\chi)$, at no stage an extension of the kind in Case 1, Subcase A, is used for inserting new rule applications above a leaf $\chi_j = \chi_j$, since otherwise the situation would occur, that $j+1 = m$ and that $i \in \{0, 1, \ldots, j-1\}$ existed with $\chi_i \equiv_{\text{ren}} \chi_j$, which would entail $\text{min-dp}_\alpha(\chi_i) = \text{min-dp}_\alpha(\chi_j)$ by (7.60) in contradiction to $\text{min-dp}_\alpha(\chi_j) < \text{min-dp}_\alpha(\chi_i)$ due to the construction of the thread.

(7.63) implies that $\alpha \in \text{fv}(\chi_m)$ and $\text{min-dp}_\alpha(\chi_m) = 0$ for the topmost formula χ_m in the thread holds and hence that $\chi_m \equiv \alpha$. Since the process P never extends a derivation above a leaf carrying the equation $\alpha = \alpha$, this means that the topmost formula in the thread postulated by (7.63) is actually itself a topmost formula in $\mathcal{D}$, which—since such a formula is unable to be bound by an application of ARROW/FIX—must be an axiom $\alpha = \alpha$ (REFL) at the top of the prooftree $\mathcal{D}$.

Hence we have shown the desired statement (7.55).

□

For being eventually able to give our proof for Lemma 7.2.6, we need also a statement that is concerned with substituting two, possibly different, recursive types σ_1 and σ_2 for some type variable α on the left-hand side, and respectively on the right-hand side of each equation between recursive types in a derivation in $\mathbf{HB}_{\mathbf{0}}^{=}$.

For formulating such a statement, we first define, under suitable restrictions, prooftrees that are the outcomes of such 'asymmetric' substitutions in $\mathbf{HB}_{\mathbf{0}}^{=}$-derivations. And in the subsequent lemma we show that, under the same conditions, these defined prooftrees are again $\mathbf{HB}_{\mathbf{0}}^{=}$-derivations.

Definition 7.2.12 ('Asymmetric' substitution in a $\mathbf{HB}_{\mathbf{0}}^{=}$-derivation). Let recursive types $\tau_1, \tau_2, \sigma_1, \sigma_2 \in \mu Tp$, a type variable $\alpha \in TVar$, and a derivation $\mathcal{D}$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ of the form

$$\begin{array}{c} \left[\dfrac{\text{(REFL)}}{\alpha = \alpha} \right] \\ \mathcal{D} \\ \tau_1 = \tau_2 \end{array} \tag{7.64}$$

be given, where the class of occurrences of axioms (REFL) of the form $\alpha = \alpha$, indicated at the top of the symbolic prooftree (7.64), are meant to gather all occurrences of axioms (REFL) of the form $\alpha = \alpha$ at the top of $\mathcal{D}$, such that

(i) for all axioms (REFL) of the form $\chi = \chi$, where $\chi \in \mu Tp$, that occur at the top of the derivation $\mathcal{D}$, either $\alpha \notin \text{fv}(\chi)$ or $\chi \equiv \alpha$ holds (in the latter case the axiom $\chi = \chi$ is then part of the class of axioms $\alpha = \alpha$ indicated at the top of the symbolic prooftree (7.64)).

(ii) for all equations between recursive types $\chi_1 = \chi_2$ that occur in $\mathcal{D}$, it holds, for all $i, j \in \{1, 2\}$, that σ_i is substitutible for α in χ_j.

Let now u be an assumption marker that does not occur in $\mathcal{D}$. Then we define by $\mathcal{D}[[\,\sigma_1/\alpha\,||\,\sigma_2/\alpha\,]]^{(u)}$ the prooftree that is of the symbolic form

$$\begin{array}{c} \big[\sigma_1 = \sigma_2\big]^u \\ \mathcal{D}[[\,\sigma_1/\alpha\,||\,\sigma_2/\alpha\,]] \\ \tau_1[\sigma_1/\alpha] = \tau_2[\sigma_2/\alpha] \end{array} \tag{7.65}$$

and that arises from the derivation $\mathcal{D}$ by

- replacing each occurrence of an equation between recursive types $\chi_1 = \chi_2$ in $\mathcal{D}$ by an occurrence of the formula $\chi_1[\sigma_1/\alpha] = \chi_2[\sigma_2/\alpha]$, and by
- attaching the assumption marker u to such equations $\sigma_1 = \sigma_2$ that are the result of substituting σ_1 and σ_2 for α in axioms (REFL) of the form $\alpha = \alpha$ at the top of $\mathcal{D}$.

That is, $\mathcal{D}[[\,\sigma_1/\alpha\,||\,\sigma_2/\alpha\,]]^{(u)}$ is the result of substituting σ_1 for α on the left-hand side and of substituting σ_2 for α on the right-hand side of each occurrence of an equation between recursive types $\chi_1 = \chi_2$ in $\mathcal{D}$, and of marking with the marker u all those assumptions $\sigma_1 = \sigma_2$ that arise by such substitutions in axiom occurrences of $\alpha = \alpha$ at the top of $\mathcal{D}$. ⊠

Lemma 7.2.13 ('Asymmetric' substitution in an HB$_\mathbf{0}^=$-derivation). *Let recursive types $\tau_1, \tau_2, \sigma_1, \sigma_2 \in \mu Tp$, a type variable $\alpha \in TVar$, and a derivation $\mathcal{D}$ in* **HB$_\mathbf{0}^=$** *of the form* (7.64) *be given such that the conditions (i) and (ii) in Definition 7.2.12 are fulfilled. Furthermore let u be an assumption marker that does not occur in $\mathcal{D}$.*

Then the prooftree $\mathcal{D}[[\,\sigma_1/\alpha\,||\,\sigma_2/\alpha\,]]^{(u)}$ is a derivation in **HB$_\mathbf{0}^=$** *that has the conclusion $\tau_1[\sigma_1/\alpha] = \tau_2[\sigma_2/\alpha]$. The derivation $\mathcal{D}[[\,\sigma_1/\alpha\,||\,\sigma_2/\alpha\,]]^{(u)}$ has the same depth and the same size as $\mathcal{D}$. And the following holds for the open marked assumptions of $\mathcal{D}[[\,\sigma_1/\alpha\,||\,\sigma_2/\alpha\,]]^{(u)}$: if the derivation $\mathcal{D}$ contains at least one occurrence of an axiom* (REFL) *in a leaf at its top, then for the open marked assumptions of $\mathcal{D}[[\,\sigma_1/\alpha\,||\,\sigma_2/\alpha\,]]^{(u)}$ it holds*

$$\begin{aligned} \mathsf{massm}(\mathcal{D}[[\,\sigma_1/\alpha\,||\,\sigma_2/\alpha\,]]^{(u)}) = \big\{(\sigma_1 = \sigma_2)^u\big\} \cup \\ \cup\ \big\{\,(\chi_1[\sigma_1/\alpha] = \chi_2[\sigma_2/\alpha])^v \mid (\chi_1 = \chi_2)^v \in \mathsf{massm}(\mathcal{D})\big\}\ ; \end{aligned}$$

and otherwise just

$$\begin{aligned} \mathsf{massm}(\mathcal{D}[[\,\sigma_1/\alpha\,||\,\sigma_2/\alpha\,]]^{(u)}) = \\ = \big\{\,(\chi_1[\sigma_1/\alpha] = \chi_2[\sigma_2/\alpha])^v \mid (\chi_1 = \chi_2)^v \in \mathsf{massm}(\mathcal{D})\big\} \end{aligned}$$

holds. If, in particular, the derivation $\mathcal{D}$ is such that it does not contain open assumptions and such that it contains at least one occurrence of an axiom (REFL) *of the form $\alpha = \alpha$ in a leaf at the top, then $\mathcal{D}[[\,\sigma_1/\alpha\,||\,\sigma_2/\alpha\,]]^{(u)}$ is a derivation in* **HB$_\mathbf{0}^=$** *with conclusion $\tau_1[\sigma_1/\alpha] = \tau_2[\sigma_2/\alpha]$ and with single open assumption class $[\sigma_1 = \sigma_2]^u$, which is inhabited.*

Sketch of Proof. The proof of this lemma consists in a straightforward induction on $|\mathcal{D}|$ and is in principle largely analogous to the proof of Lemma 7.1.4. □

The only missing element for finally concluding the proof of Theorem 7.2.1 consists in a proof of Lemma 7.2.6. But such a proof can now be assembled easily with the help of Lemma 7.2.9 and Lemma 7.2.13.

Proof of Lemma 7.2.6. Let arbitrary $\tau \in \mu Tp$ and $\alpha \in TVar$ with the property $\alpha \downarrow \tau$ and $\alpha \in \mathrm{fv}(\tau)$ be given. We have to show that a derivation $\mathcal{D}_{\tau,\alpha;\sigma_1,\sigma_2}$ of the form (7.37) and with single inhabited open assumption class $[\sigma_1 = \sigma_2]^u$, for some assumption marker u, can effectively be found.

By Lemma 7.2.9 a derivation $\mathcal{D}_{\tau,\alpha}^{(\sigma_1,\sigma_2)}$ of the form (7.38) without open assumption classes can effectively be found, which with respect to σ_1 and σ_2 satisfies the assumptions of Lemma 7.2.13 (i.e. that $\mathcal{D}_{\tau,\alpha}^{(\sigma_1,\sigma_2)}$ is of the form with the conditions (i) and (ii) in (7.64) Definition 7.2.12 fulfilled); we choose $\mathcal{D}_{\tau,\alpha}^{(\sigma_1,\sigma_2)}$ in this way. Hence Lemma 7.2.13 can be applied and it gives that, for some assumption markers u, the derivation $\mathcal{D}_{\tau,\alpha}^{(\sigma_1,\sigma_2)}[[\sigma_1/\alpha||\sigma_2/\alpha]]^{(u)}$ is a derivation in $\mathbf{HB}_{\mathbf{0}}^{=}$ of the form (7.37), and that this derivation contains the single open undischarged assumption class $[\sigma_1 = \sigma_2]^u$, which is inhabited since the class of axioms $[\alpha = \alpha]$ is inhabited by the choice of $\mathcal{D}_{\tau,\alpha}^{(\sigma_1,\sigma_2)}$. This shows the claim of Lemma 7.2.6.

□ □

We conclude this section by giving a very easy example for the effective transformation, asserted by Theorem 7.2.1 and described in the proofs given here, from derivations in $\mathbf{AC}^{=}$ or in $\mathbf{AC}^{=}_{*}$ without assumptions into derivations in $\mathbf{HB}^{=}$ with respectively the same conclusion and without open assumption classes.

Example 7.2.14 (Transforming an $\mathbf{AC}^{=}_{*}$-derivation into a $\mathbf{HB}^{=}$-derivation). We consider the derivation $\mathcal{D}$ in $\mathbf{AC}^{=}_{*}$ given in Example 5.1.7 for the equation between recursive types $\tau_1 = \sigma_1$ with $\sigma_1 =_{\mathrm{def}} \mu\beta.\,((\beta \to \bot) \to \bot)$ and $\tau_1 =_{\mathrm{def}} \mu\alpha.\,(\alpha \to \bot)$ and that with respect to its subderivation $\mathcal{D}_1$ of the form

$$\dfrac{\begin{array}{c}(\text{FOLD/UNFOLD})\\ \tau_1 = \tau_1 \to \bot\end{array} \qquad \dfrac{\dfrac{(\text{FOLD/UNFOLD})}{\tau_1 = \tau_1 \to \bot} \qquad \dfrac{(\text{REFL})}{\bot = \bot}}{\tau_1 \to \bot = (\tau_1 \to \bot) \to \bot}\ \text{ARROW}}{\tau_1 = (\tau_1 \to \bot) \to \bot}\ \text{TRANS}$$

can be written as

$$\dfrac{\begin{array}{c}\mathcal{D}_1\\ \tau_1 = \underbrace{(\tau_1 \to \bot) \to \bot}_{\equiv ((\beta\to\bot)\to\bot)\,[\tau_1/\beta]}\end{array}}{\underbrace{\tau_1}_{\equiv \mu\alpha.\,(\alpha\to\bot)} \quad = \quad \underbrace{\sigma_1}_{\equiv \mu\beta.\,((\beta\to\bot)\to\bot)}}\ \text{CONTRACT} \tag{7.66}$$

We are going to transform the $\mathbf{AC}^{=}_{*}$-derivation $\mathcal{D}$, according to the transformation described in the proof of Theorem 7.2.1, and thereby in the proofs of the lemmas used for this theorem, into a derivation $\mathcal{D}'$ in $\mathbf{HB}^{=}$ with the same conclusion and without open assumptions. In doing so we will perform the three steps of this transformation that are diagrammatically outlined in Figure 7.2.

For the first step of the transformation, there is no work to be done here. This is because $\mathcal{D}$ does not contain applications of μ-COMPAT nor occurrences of the axiom $(\mu{-}\bot)$, and hence $\mathcal{D}$ is also a derivation without assumptions in the system $\mathbf{AC}^{=}_{-}$. Therefore, the first step in the transformation described in the proof of Theorem 7.2.1, elimination of μ-COMPAT-applications in $\mathcal{D}$, is unnecessary here, and we may take $\mathcal{D}$ itself for the outcome of this trivial first step, the derivation $\mathcal{D}^{(1)}$ in $\mathbf{AC}^{=}_{-}$ that mimics $\mathcal{D}$.

In the second step of the transformation described into the proof of Theorem 7.2.1, $\mathcal{D}^{(1)}$ is translated in a derivation in $\mathbf{HB}^{=} + \mathrm{CONTRACT}^{-}_{(\mathrm{nd})}$ by simply changing the rule labels of applications of CONTRACT into $\mathrm{CONTRACT}^{-}_{(\mathrm{nd})}$. Here we arrive at a derivation $\mathcal{D}^{(2)}$ of the form

$$\dfrac{\dfrac{\dfrac{\text{(FOLD/UNFOLD)}}{\tau_1 = \tau_1 \to \bot} \qquad \dfrac{\dfrac{\text{(FOLD/UNFOLD)}}{\tau_1 = \tau_1 \to \bot} \quad \dfrac{\text{(REFL)}}{\bot = \bot}}{\tau_1 \to \bot = (\tau_1 \to \bot) \to \bot}\,\text{ARROW}}{\tau_1 = (\tau_1 \to \bot) \to \bot}\,\text{TRANS}}{\tau_1 = \sigma_1}\,\mathrm{CONTRACT}^{-}_{(\mathrm{nd})} \tag{7.67}$$

with just one application of $\mathrm{CONTRACT}^{-}_{(\mathrm{nd})}$. Since all of the axioms and rules occurring in it are also part of $\mathbf{HB}^{=}$, the subderivation $\mathcal{D}^{(2)}_1$ leading up to the premise of the application of $\mathrm{CONTRACT}^{-}_{(\mathrm{nd})}$ in $\mathcal{D}^{(2)}$ is already a derivation in $\mathbf{HB}^{=}$.

In the third step of the transformation, we are therefore only required to eliminate the application ι of $\mathrm{CONTRACT}^{-}_{(\mathrm{nd})}$ at the bottom of $\mathcal{D}^{(2)}$, and we can do so by an appeal to the transformation stated by Lemma 7.2.3, (i). From the proof of this lemma we learn that for building up the translation of the application ι of $\mathrm{CONTRACT}^{-}_{(\mathrm{nd})}$ into the mimicking derivation context $\mathcal{DC}^{(\iota)}_{\mathrm{mim}}$ in $\mathbf{HB}^{=}$, we need to obtain a derivation $\mathcal{D}_{(\beta\to\bot)\to\bot,\,\beta;\,\tau_1,\,\sigma_1}$ which, due to $\beta \in \mathrm{fv}((\beta \to \bot) \to \bot)$ and $\beta \downarrow (\beta \to \bot) \to \bot$, is guaranteed to exist by Lemma 7.2.6. And for producing $\mathcal{D}_{(\beta\to\bot)\to\bot,\,\beta;\,\tau_1,\,\sigma_1}$ in its turn, we see from the proof of Lemma 7.2.6 that also a derivation $\mathcal{D}^{(\tau_1,\,\sigma_1)}_{(\beta\to\bot)\to\bot,\,\beta}$ with the properties as stated by Lemma 7.2.9 is needed.

In carrying out the procedure detailed in the proof of Lemma 7.2.9 for building up the $\mathbf{HB}^{=}_{\mathbf{0}}$-derivation $\mathcal{D}^{(\tau_1,\,\sigma_1)}_{(\beta\to\bot)\to\bot,\,\beta}$, we arrive at the derivation

$$\dfrac{\dfrac{\dfrac{\text{(REFL)}}{\beta = \beta} \quad \dfrac{\text{(REFL)}}{\bot = \bot}}{\beta \to \bot = \beta \to \bot}\,\text{ARROW} \qquad \dfrac{\text{(REFL)}}{\bot = \bot}}{(\beta \to \bot) \to \bot = (\beta \to \bot) \to \bot}\,\text{ARROW}$$

that does not contain open assumptions (nor for that matter discharged assumptions). The derivation $\mathcal{D}^{(\tau_1,\,\sigma_1)}_{\tau,\,\beta}$, where $\tau =_{\text{def}} (\beta \to \bot) \to \bot$, is seemingly just a very redundant derivation in $\mathbf{HB}^{=}_{\mathbf{0}}$, and here also in $\mathbf{HB}^{=}$, for its conclusion $\tau = \tau$ that corresponds to an axiom (REFL). However, the equation $\tau = \tau$ is formally proven by $\mathcal{D}^{(\tau_1,\,\sigma_2)}_{\tau,\,\beta}$ in an 'analytical way' and such that β occurs in an axiom $\chi = \chi$ at the top only if $\chi \equiv \beta$. This makes it possible to mimic inferences in $\mathbf{AC}^{=}_{*}$ or $\mathbf{AC}^{=}_{*-}$ involving substitutions of recursive types for β in τ (such as the application of $\text{CONTRACT}^{-}_{(\text{nd})}$ at the bottom of the derivation $\mathcal{D}^{(2)}$ in (7.67)) by derivations in $\mathbf{HB}^{=}_{\mathbf{0}}$, and eventually be derivations in $\mathbf{HB}^{=}$.

As in the proof of Lemma 7.2.6, the derivation $\mathcal{D}_{(\beta\to\bot)\to\bot,\,\beta;\,\tau_1,\,\sigma_1}$ can now be chosen as $\mathcal{D}^{(\tau_1,\,\sigma_1)}_{(\beta\to\bot)\to\bot,\,\beta}[[\tau_1/\beta||\sigma_1/\beta]]^{(u)}$, i.e. as the result of substituting τ_1 for β on the left-hand side and of substituting σ_1 for β on the right-hand side of every formula in $\mathcal{D}^{(\tau_1,\,\sigma_1)}_{(\beta\to\bot)\to\bot,\,\beta}$, and of attaching the assumption marker u to those formula occurrences of $\tau_1 = \sigma_1$ at the top that arise by such substitutions in axiom occurrences of $\alpha = \alpha$. In this way we find here the derivation $\mathcal{D}_{(\beta\to\bot)\to\bot,\,\beta;\,\tau_1,\,\sigma_1}$ of the form

$$\dfrac{\dfrac{(\tau_1 = \sigma_1)^u \qquad \dfrac{\text{(REFL)}}{\bot = \bot}}{\tau_1 \to \bot = \sigma_1 \to \bot}\,\text{ARROW} \qquad \dfrac{\text{(REFL)}}{\bot = \bot}}{(\tau_1 \to \bot) \to \bot = (\sigma_1 \to \bot) \to \bot}\,\text{ARROW}$$

with the single open assumption $(\tau_1 = \sigma_1)^u$.

Using this derivation, we can now construct, according to Figure 7.3, the unary mimicking derivation-context $\tilde{\mathcal{DC}}^{(\iota)}_{\text{mim}}$ in $\mathbf{HB}^{=}_{\mathbf{0}}$+TRANS for the application ι of $\text{CONTRACT}^{-}_{(\text{nd})}$ at the bottom of $\mathcal{D}^{(2)}$ where $\tilde{\mathcal{DC}}^{(\iota)}_{\text{mim}}$ is now of the form

$$\dfrac{\big[\,\tau_1 = (\tau_1 \to \bot) \to \bot\,\big]_1 \qquad \text{FOLD}_r\,\dfrac{\text{ARROW/FIX},\,\boldsymbol{v}\,\dfrac{\text{ARROW}\,\dfrac{\dfrac{\big[\,\tau_1 = (\tau_1 \to \bot) \to \bot\,\big]_1 \qquad \dfrac{\big((\tau_1 \to \bot) \to \bot = (\sigma_1 \to \bot) \to \bot\big)^{\boldsymbol{v}}}{(\tau_1 \to \bot) \to \bot = \sigma_1}\,\text{FOLD}_r}{\tau_1 = \sigma_1}\,\text{TRANS} \qquad \dfrac{\text{(REFL)}}{\bot = \bot}}{\tau_1 \to \bot = \sigma_1 \to \bot} \qquad \dfrac{\text{(REFL)}}{\bot = \bot}}{(\tau_1 \to \bot) \to \bot = (\sigma_1 \to \bot) \to \bot}}{(\tau_1 \to \bot) \to \bot = \sigma_1}}{\tau_1 = \sigma_1}\,\text{TRANS}$$

(as in Figure 7.3 we have again indicated for reading convenience the premise of ι, namely $\tau_1 = (\tau_1 \to \bot) \to \bot$, inside occurrences of the context-hole $[\,]_1$ in $\tilde{\mathcal{DC}}^{(\iota)}_{\text{mim}}$). And by inserting the immediate subderivation $\mathcal{D}^{(2)}_1$ of $\mathcal{D}^{(2)}$ in (7.67) into occurrences of the context-holes $[\,]_1$ of $\tilde{\mathcal{DC}}^{(\iota)}_{\text{mim}}$, we can now build the derivation $\tilde{\mathcal{D}}'$ of the form

$$\dfrac{\begin{array}{c}\mathcal{D}_1\\ \tau_1 = (\tau_1 \to \bot) \to \bot\end{array} \qquad \text{FOLD}_r\,\dfrac{\text{ARROW/FIX}, \boldsymbol{u}\;\dfrac{\text{ARROW}\;\dfrac{\dfrac{\begin{array}{c}\mathcal{D}_1\\ \tau_1 = (\tau_1 \to \bot) \to \bot\end{array} \qquad \dfrac{\big((\tau_1 \to \bot) \to \bot = (\sigma_1 \to \bot) \to \bot\big)^{\boldsymbol{u}}}{(\tau_1 \to \bot) \to \bot = \sigma_1}\,\text{FOLD}_r}{\tau_1 = \sigma_1}\,\text{TRANS} \qquad \dfrac{\text{(REFL)}}{\bot = \bot}}{\tau_1 \to \bot = \sigma_1 \to \bot} \qquad \dfrac{\text{(REFL)}}{\bot = \bot}}{(\tau_1 \to \bot) \to \bot = (\sigma_1 \to \bot) \to \bot}}{(\tau_1 \to \bot) \to \bot = \sigma_1}}{\tau_1 = \sigma_1}\,\text{TRANS}$$

in $\mathbf{HB}^{=} + \text{TRANS} + \text{FOLD}_r$ that contains no open assumptions and has the same conclusion as $\mathcal{D}$. By eliminating from $\tilde{\mathcal{D}}'$ the two applications of FOLD_r in $\tilde{\mathcal{D}}'$ in an obvious way (according to the transformation described in the proof of Lemma 5.1.19), we can finally build a derivation $\mathcal{D}'$ in $\mathbf{HB}^{=}$ without open assumptions and with the same conclusion $\tau_1 = \sigma_1$ as $\mathcal{D}$. We only indicate that the (subderivation belonging to) topmost application of FOLD_r in $\tilde{\mathcal{D}}'$ can be replaced by the mimicking derivation

$$\dfrac{\big((\tau_1 \to \bot) \to \bot = (\sigma_1 \to \bot) \to \bot\big)^{\boldsymbol{v}} \qquad \dfrac{\dfrac{\text{(FOLD/UNFOLD)}}{\sigma_1 = (\sigma_1 \to \bot) \to \bot}}{(\sigma_1 \to \bot) \to \bot = \sigma_1}\,\text{SYMM}}{(\tau_1 \to \bot) \to \bot = \sigma_1}\,\text{TRANS}$$

in $\mathbf{HB}^{=}$ and do not typeset the resulting derivation $\mathcal{D}'$ here due to its typographical breadth.

In this way we have carried out the three steps of the transformation and have arrived at a derivation $\mathcal{D}'$ in $\mathbf{HB}^{=}$ without open assumptions and with the same conclusion $\tau_1 = \sigma_1$ as the derivation $\mathcal{D}$ in $\mathbf{AC}^{=}_{*}$ from which we started.

We conclude this example by observing that the derivation $\mathcal{D}'$ in $\mathbf{HB}^{=}$ that we have found here is in fact not of mimimal size for a derivation in $\mathbf{HB}^{=}$ with conclusion $\tau_1 = \sigma_1$. Indeed, it is a little more complex than a derivation $\tilde{\mathcal{D}}_2$ in $\mathbf{HB}^{=}$ without open assumptions and with the same conclusion as $\mathcal{D}'$ that results from the $\mathbf{HB}^{=}_{\mathbf{0}}$-derivation $\mathcal{D}_2$ in Figure 6.8 by transforming it into a derivation in $\mathbf{HB}^{=}$: $\mathcal{D}'$ has depth $|\mathcal{D}'| = 7$ and size $\text{s}(\mathcal{D}') = 23$, whereas $\tilde{\mathcal{D}}_2$ has depth[7] $|\tilde{\mathcal{D}}_2| = 7$ and size $\text{s}(\tilde{\mathcal{D}}_2) = 16$. ■

The observation at the end of the example above indicates that the transformation developed in this section does not necessarily produce mimicking derivations in $\mathbf{HB}^{=}$ of minimal size, nor of minimal depth (as can be seen by other examples), for given derivations in $\mathbf{AC}^{=}$ and $\mathbf{AC}^{=}_{*}$ without assumptions.

[7] More precisely, $|\tilde{\mathcal{D}}_2| = 7$ can be reached by replacing from the two successive applications of $\text{FOLD}_{l/r}$ at the top of $\mathcal{D}$ the application of FOLD_l first, and then the application of FOLD_r (for which an additional application of SYMM is needed in a mimicking derivation).

Chapter 8

Transforming Derivations from $\mathbf{HB}^=$ to $\mathbf{AC}^=$

In this chapter we will develop, in two stages, an effective transformation from derivations without open assumption classes in the Brandt-Henglein system $\mathbf{HB}^=$ into derivations with respectively the same conclusion in the system $\mathbf{AC}^=$ of Amadio and Cardelli. The core of this transformation consists of a method, described in Section 8.1, of building, for every derivation without open assumption classes in the variant-Brandt-Henglein system $\mathbf{HB}_\mathbf{0}^=$, a derivation in $\mathbf{AC}^=$ with the same conclusion and without assumptions. Unfortunately, this method cannot be generalized, at least not directly, to one that defined a similar transformation from arbitrary derivations in the original Brandt-Henglein system $\mathbf{HB}^=$ into derivations in $\mathbf{AC}^=$; however, it can be adapted to $\mathbf{HB}^=$-derivations that arise[1] from $\mathbf{HB}_\mathbf{0}^=$-derivations. But instead of generalizing the transformation between $\mathbf{HB}_\mathbf{0}^=$- and $\mathbf{AC}^=$-derivations to derivations in $\mathbf{HB}^=$, we will complement it in Section 8.2 with an effective method that accomplishes the following task: transforming an arbitrary derivation in $\mathbf{HB}^=$ without open assumptions by a 'symmetry-and-transitivity-elimination procedure' into a derivation in $\mathbf{HB}_\mathbf{0}^=$ with the same conclusion and without open assumptions.

Hereby perhaps only some rather more theoretical value can be attributed to the second part, the transformation developed in Section 8.2 from $\mathbf{HB}^=$-derivations without open assumptions into mimicking $\mathbf{HB}_\mathbf{0}^=$-derivations. This is because of the following two facts. Firstly, if an equation $\tau = \sigma$, for some $\tau, \sigma \in \mu Tp$, is provable by a derivation in $\mathbf{HB}^=$ without open assumptions, then τ and σ are strongly equiv-

[1] By this we mean derivations in $\mathbf{HB}^=$ that are the 'image' of $\mathbf{HB}_\mathbf{0}^=$-derivations under the transformation described in the proof of Lemma 5.1.19. A derivation $\mathcal{D}'$ in $\mathbf{HB}^=$ that is the result of applying the procedure implicit in the proof of Lemma 5.1.19 to a derivation in $\mathcal{D}$ in $\mathbf{HB}_\mathbf{0}^=$ is of a *special form*: each application of SYMM in $\mathcal{D}$ occurs immediately below an axiom (FOLD/UNFOLD) or $(\mu{-}\bot)'$, and each application of TRANS in $\mathcal{D}'$ has an immediate subderivation that terminates with an application of SYMM, or that consists of an axiom (FOLD/UNFOLD) or $(\mu{-}\bot)'$.

alent due to the completeness of $\mathbf{HB}^{=}$ with respect to $=_{\mu}$. And secondly, for every pair $\langle\tau,\sigma\rangle$ of strongly equivalent recursive types, a derivation in $\mathbf{HB}_{\mathbf{0}}^{=}$ without open assumptions and with $\tau=\sigma$ as its conclusion can effectively be produced, by an algorithmic method that can be extracted from the completeness proof for $\mathbf{HB}^{=}$ by Brandt and Henglein in [BrHe98]. However, the transformation from $\mathbf{HB}^{=}$ to $\mathbf{HB}_{\mathbf{0}}^{=}$ described in Section 8.2 is interesting in its own right: it can be viewed as a method of 'normalizing' $\mathbf{HB}^{=}$-derivations with the outcome of $\mathbf{HB}_{\mathbf{0}}^{=}$-derivations of a particular form[2] that correspond to $\mathbf{HB}^{=}$-derivations very directly. This transformation proceeds, in a manner reminiscent of classical cut-elimination procedures, by stepwisely upwards-permuting applications of the symmetry rule, and separately, of applications of the transitivity rule in derivations of $\mathbf{HB}_{\mathbf{0}}^{=}+\mathrm{SYMM}+\mathrm{TRANS}$, and by ultimately eliminating such rule applications altogether.

The following theorem gathers the main result of the two sections of this chapter about effective transformations of $\mathbf{HB}_{\mathbf{0}}^{=}$-derivations into $\mathbf{AC}^{=}$-derivations, and of $\mathbf{HB}^{=}$-derivations into $\mathbf{HB}_{\mathbf{0}}^{=}$-derivations, for stating the existence of an effective transformation of $\mathbf{HB}_{\mathbf{0}}^{=}$- into $\mathbf{AC}^{=}$-derivations.

Theorem 8.0.1 (Effective transformation of $\mathbf{HB}^{=}$- into $\mathbf{AC}^{=}$-derivations). *Every derivation $\mathcal{D}$ in the system $\mathbf{HB}^{=}$ with conclusion $\tau=\sigma$ (for arbitrary $\tau,\sigma\in\mu Tp$) and without open assumption classes can be transformed by a sequence of effective steps into a derivation $\mathcal{D}'$ in the system $\mathbf{AC}^{=}$ with the same conclusion.*

Proof. This theorem is an immediate consequence of the main results of the two subsections of this section, of Theorem 8.2.2 on page 284 and of Theorem 8.1.8 on page 277. □

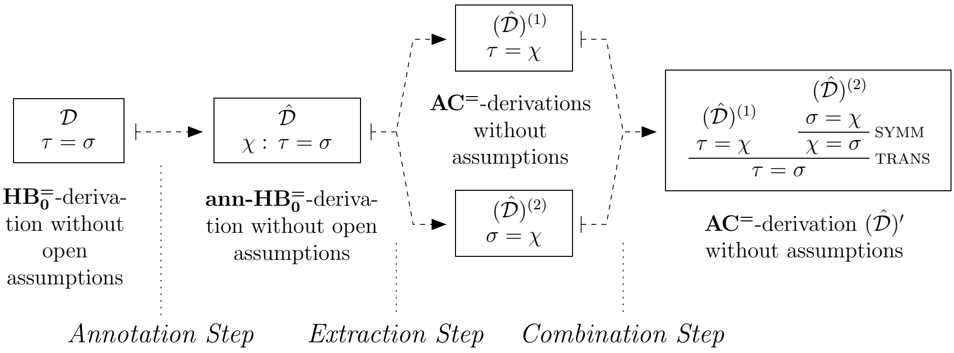
8.1 A Transformation of Derivations in $\mathbf{HB}_{\mathbf{0}}^{=}$ into Derivations in $\mathbf{AC}^{=}$

In this section an effective transformation of derivations in the Brandt-Henglein system $\mathbf{HB}_{\mathbf{0}}^{=}$ without open assumptions into derivations in the Amadio-Cardelli system $\mathbf{AC}^{=}$ will be developed. As an important auxiliary concept for this transformation, we will introduce an annotated version $\textbf{ann-HB}_{\mathbf{0}}^{=}$ of the proof system $\mathbf{HB}_{\mathbf{0}}^{=}$. Formulas of the system $\textbf{ann-HB}_{\mathbf{0}}^{=}$ are equations between recursive types that are additionally annotated by recursive types. The role being played by the annotated system $\textbf{ann-HB}_{\mathbf{0}}^{=}$ for the transformation to develop here consists in facilitating the 'extraction' of an $\mathbf{AC}^{=}$-derivation from a $\mathbf{HB}_{\mathbf{0}}^{=}$-derivation.

Our transformation will namely first annotate a given derivation $\mathcal{D}$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ without open assumptions, producing the intermediary result of a derivation $\hat{\mathcal{D}}$ in $\textbf{ann-HB}_{\mathbf{0}}^{=}$ without open assumptions, before transforming $\hat{\mathcal{D}}$ further into a derivation $(\hat{\mathcal{D}})'$ in $\mathbf{AC}^{=}$ without assumptions and with the same conclusion as $\mathcal{D}$. Spelt

[2] Cf. the description of $\mathbf{HB}^{=}$-derivations of 'special form' in footnote 1.

Figure 8.1: Illustration of the main steps in the transformation developed in this section that for an arbitrary derivation in $\mathbf{HB_0^=}$ without open assumptions is able to construct a derivation in $\mathbf{AC^=}$ with the same conclusion and without assumptions.


$\mathcal{D}$
$\tau = \sigma$
$\mathbf{HB_0^=}$-derivation without open assumptions
$\hat{\mathcal{D}}$
$\chi : \tau = \sigma$
ann-$\mathbf{HB_0^=}$-derivation without open assumptions
$(\hat{\mathcal{D}})^{(1)}$
$\tau = \chi$
$\mathbf{AC^=}$-derivations without assumptions
$(\hat{\mathcal{D}})^{(2)}$
$\sigma = \chi$
$(\hat{\mathcal{D}})^{(1)}$
$\tau = \chi$
$(\hat{\mathcal{D}})^{(2)}$
$\sigma = \chi$
SYMM
$\chi = \sigma$
TRANS
$\tau = \sigma$
$\mathbf{AC^=}$-derivation $(\hat{\mathcal{D}})'$ without assumptions
Annotation Step
Extraction Step
Combination Step

out in a little more details, the transformation will proceed, starting from an arbitrary derivation $\mathcal{D}$ in $\mathbf{HB_0^=}$ without open assumptions and with conclusion $\tau = \sigma$ (for some $\tau, \sigma \in \mu Tp$) by performing the following three steps:

(1) *Annotation Step*: The derivation $\mathcal{D}$ in $\mathbf{HB_0^=}$ is annotated with recursive types such that the result is a derivation $\hat{\mathcal{D}}$ in **ann-**$\mathbf{HB_0^=}$ with conclusion $\chi : \tau = \sigma$, for some annotation $\chi \in \mu Tp$, and equally as $\mathcal{D}$ without open assumption classes. In this step the structure of the derivation $\mathcal{D}$ gets 'analyzed', to some extent, by the annotation χ in the conclusion of $\hat{\mathcal{D}}$. The underlying process of annotating $\mathbf{HB_0^=}$-derivations is rather straightforward, proceeds by finding appropriate annotations for the occurring formulas, and can be defined by induction on the depth of $\mathbf{HB_0^=}$-derivations.

(2) *Extraction Step*: From the derivation $\hat{\mathcal{D}}$ in **ann-**$\mathbf{HB_0^=}$ that has the conclusion $\chi : \tau = \sigma$, but that does not have open assumptions, two derivations $(\hat{\mathcal{D}})^{(1)}$ and $(\hat{\mathcal{D}})^{(2)}$ in $\mathbf{AC^=}$ with respective conclusions $\tau = \chi$ and $\sigma = \chi$ and without assumptions are 'extracted'. The process underlying the extraction of two such $\mathbf{AC^=}$-derivations from an **ann-**$\mathbf{HB_0^=}$-derivation can again be defined by induction on the depth of **ann-**$\mathbf{HB_0^=}$-derivations.

(3) *Combination Step*: The derivations $(\hat{\mathcal{D}})^{(1)}$ and $(\hat{\mathcal{D}})^{(2)}$ in $\mathbf{AC^=}$ with the respective conclusions $\tau = \chi$ and $\sigma = \chi$ are combined by one additional application of each of the rules SYMM and TRANS into an $\mathbf{AC^=}$-derivation $(\hat{\mathcal{D}})'$ without assumptions and with the same conclusion $\tau = \sigma$ as $\mathcal{D}$.

An illustration of these three steps is given in Figure 8.1. The justification for the transformation steps (1), (2), and (3) will be provided below by the proofs of Lemma 8.1.5, (i), of Lemma 8.1.6, (i), and of Lemma 8.1.6, (ii), respectively.

There is, however, one additional complication of the transformation developed below that has not been mentioned in the outline just given. Namely, the transformation justified by Lemma 8.1.6, (i), will actually proceed by (1) extracting two derivations in the μ-COMPAT-free variant system $\mathbf{AC}^{=}_{*-}$ of the system $\mathbf{AC}^{=}_{*}$, which differs from $\mathbf{AC}^{=}$ by the absence of the rule UFP and the presence instead of the rule CONTRACT, and by then (2) translating these two $\mathbf{AC}^{=}_{*-}$-derivations into respective mimicking derivations in $\mathbf{AC}^{=}$. The reason for this is that it turns out to be slightly more immediate to extract, for arbitrary derivations $\hat{\mathcal{D}}$ in $\mathbf{ann\text{-}HB}^{=}_{\mathbf{0}}$ without open assumptions, two derivations $(\hat{\mathcal{D}})^{(1)}_{*-}$ and $(\hat{\mathcal{D}})^{(2)}_{*-}$ in the version $\mathbf{AC}^{=}_{*-}$ of the Amadio-Cardelli system containing applications of CONTRACT instead of derivations $(\hat{\mathcal{D}})^{(1)}$ and $(\hat{\mathcal{D}})^{(2)}$ in $\mathbf{AC}^{=}$ that have to rely on applications of UFP. As an obvious consequence of justifying the extracting step in this manner via $\mathbf{AC}^{=}_{*-}$-derivations, the results of this section lead actually to the in effect stronger statement of a transformation from derivations in $\mathbf{HB}^{=}_{\mathbf{0}}$ without open assumptions into derivations in $\mathbf{AC}^{=}_{*-}$ without assumptions and with the same respective conclusions (and similarly, to an analogous transformation with resulting $\mathbf{AC}^{=}_{-}$-derivations).

Now we start with developing the annotation step of the transformation outlined above. As a prerequisite for this step, the annotated version $\mathbf{ann\text{-}HB}^{=}_{\mathbf{0}}$ mentioned above of the variant-Brandt-Henglein system $\mathbf{HB}^{=}_{\mathbf{0}}$ is needed. Below we give the definition of this annotated system. The formulas of $\mathbf{ann\text{-}HB}^{=}_{\mathbf{0}}$ are equations between recursive types that are additionally annotated by recursive types. And the axioms and rules of $\mathbf{ann\text{-}HB}^{=}_{\mathbf{0}}$ are the result of adding appropriate annotations to corresponding axioms and rules of $\mathbf{HB}^{=}_{\mathbf{0}}$.

Definition 8.1.1 (The annotated version $\mathbf{ann\text{-}HB}^{=}_{\mathbf{0}}$ of the system $\mathbf{HB}^{=}_{\mathbf{0}}$). The natural-deduction-style proof system $\mathbf{ann\text{-}HB}^{=}_{\mathbf{0}}$ is defined as follows.

The *formulas* of $\mathbf{ann\text{-}HB}^{=}_{\mathbf{0}}$ are precisely all recursive-type annotated equations of recursive types, i.e. all formal objects of the form $\chi : \tau = \sigma$ with $\chi, \tau, \sigma \in \mu Tp$. The *axioms* of $\mathbf{ann\text{-}HB}^{=}_{\mathbf{0}}$ are exactly all those formulas of $\mathbf{ann\text{-}HB}^{=}_{\mathbf{0}}$ that belong to the formula scheme (REFL) shown in Figure 8.2. The *marked assumptions* that may be used in derivations in $\mathbf{ann\text{-}HB}^{=}_{\mathbf{0}}$ are, as is also shown in Figure 8.2, marked formulas of the form $(\alpha : \tau = \sigma)^{\alpha}$ with $\tau, \sigma \in \mu Tp$ and $\alpha \in TVar$ such that $\alpha \notin \mathrm{fv}(\tau) \cup \mathrm{fv}(\sigma)$.

The *inference rules* of $\mathbf{ann\text{-}HB}^{=}_{\mathbf{0}}$ are the seven rules of the seven rules REN, $(\mu\text{--}\bot)^{\bot\mathrm{der}}_{l}$, $(\mu\text{--}\bot)^{\bot\mathrm{der}}_{r}$, FOLD$_l$, FOLD$_r$, ARROW and ARROW/FIX that are are schematically defined in Figure 8.2 (to underscore the correspondence between rules of $\mathbf{ann\text{-}HB}^{=}_{\mathbf{0}}$ and rules of $\mathbf{HB}^{=}_{\mathbf{0}}$, we use the same names for corresponding rules). Notably different from the case of the underlying system $\mathbf{HB}^{=}_{\mathbf{0}}$, applications of the rules ARROW and ARROW/FIX are subject the following side-condition $\mathbf{S}$ on the annotations in open assumptions of the derivations $\mathcal{D}_1$ and $\mathcal{D}_2$ leading up to the left and, respectively, to the right premise of an ARROW or ARROW/FIX-application:

(a) *In different open assumption classes in $\mathcal{D}_1$ and $\mathcal{D}_2$ different annotation-variables occur.* Put differently and more formally, if, for some $i \in \{1, 2\}$, it holds that $(\alpha : \tau_i = \sigma_i)^{\alpha}$ is an open marked assumption in $\mathcal{D}_i$ and $(\alpha : \tau_{3-i} = \sigma_{3-i})^{\alpha}$

Figure 8.2: The annotated version $\mathbf{ann\text{-}HB}_0^=$ of the 'analytic' Brandt-Henglein system $\mathbf{HB}_0^=$.

The *axioms* and possible *marked assumptions* in $\mathbf{ann\text{-}HB}_0^=$:

$$\text{(REFL)}\quad \overline{\tau : \tau = \tau} \qquad\qquad \text{(Assm)}\quad (\alpha : \tau = \sigma)^{\alpha} \quad (\text{if } \alpha \notin \text{fv}(\tau) \cup \text{fv}(\sigma))$$

The *inference rules* of $\mathbf{ann\text{-}HB}_0^=$:

$$\frac{\chi : \tau = \sigma}{\chi : \tau' = \sigma'}\,\text{REN} \quad (\text{where } \tau' \equiv_{\text{ren}} \tau \text{ and } \sigma' \equiv_{\text{ren}} \sigma)$$

$$\frac{\chi : \bot = \sigma}{\chi : \mu\alpha\alpha_1 \ldots \alpha_n.\, \alpha = \sigma}\,(\mu\text{–}\bot)_l^{\bot\text{der}} \qquad \frac{\chi : \tau = \bot}{\chi : \tau = \mu\beta\beta_1 \ldots \beta_n.\, \beta}\,(\mu\text{–}\bot)_r^{\bot\text{der}}$$

$$\frac{\chi : \tau[\mu\alpha.\, \tau/\alpha] = \sigma}{\chi : \mu\alpha.\, \tau = \sigma}\,\text{FOLD}_l \qquad \frac{\chi : \tau = \sigma[\mu\beta.\, \sigma/\alpha]}{\chi : \tau = \mu\beta.\, \sigma}\,\text{FOLD}_r$$

$$\frac{\begin{array}{c}\mathcal{D}_1\\ \chi_1 : \tau_1 = \sigma_1\end{array} \qquad \begin{array}{c}\mathcal{D}_2\\ \chi_2 : \tau_2 = \sigma_2\end{array}}{\chi_1 \to \chi_2 : \tau_1 \to \tau_2 = \sigma_1 \to \sigma_2}\,\text{ARROW} \quad (\text{if side-cond. } \mathbf{S})$$

$$\frac{\begin{array}{c}[\alpha : \tau_1 \to \tau_2 = \sigma_1 \to \sigma_2]^{\alpha}\\ \mathcal{D}_1\\ \chi_1 : \tau_1 = \sigma_1\end{array} \qquad \begin{array}{c}[\alpha : \tau_1 \to \tau_2 = \sigma_1 \to \sigma_2]^{\alpha}\\ \mathcal{D}_2\\ \chi_2 : \tau_2 = \sigma_2\end{array}}{\mu\alpha.\, (\chi_1 \to \chi_2) : \tau_1 \to \tau_2 = \sigma_1 \to \sigma_2}\,\text{ARROW/FIX}, \alpha \quad (\text{if side-cond.s. } \mathbf{S} \text{ and } \mathbf{I})$$

is an open marked assumption in $\mathcal{D}_{3-i}$, then it must be the case that $\tau_1 \equiv \tau_2$ and $\sigma_1 \equiv \sigma_2$.

(b) *Type variables used in annotations of open marked assumptions in $\mathcal{D}_1$ do not occur free in axioms used in $\mathcal{D}_2$, and vice versa.* More precisely: if, for some $i \in \{1, 2\}$, the formula $(\alpha : \tau_i = \sigma_i)^{\alpha}$ is an open marked assumption in $\mathcal{D}_i$ and the axiom $\tau : \tau = \tau$ occurs as a leaf at the top of $\mathcal{D}_{3-i}$, then $\alpha \notin \text{fv}(\tau)$ must be satisfied.

(c) *Type variables used in annotations of open marked assumptions in $\mathcal{D}_1$ do not occur free in annotated equations occurring in open marked assumptions in $\mathcal{D}_2$, and vice versa.* This means in more detail: If, for some $i \in \{1, 2\}$, the formula $(\alpha_i : \tau_i = \sigma_i)^{\alpha}$ is an open marked assumption in $\mathcal{D}_i$ and $(\alpha_{3-i} : \tau_{3-i} = \sigma_{3-i})^{\alpha}$

is an open marked assumption in $\mathcal{D}_{3-i}$, then it follows that

$$\alpha_1 \notin \mathrm{fv}(\tau_2) \cup \mathrm{fv}(\sigma_2) \;\text{ and }\; \alpha_2 \notin \mathrm{fv}(\tau_1) \cup \mathrm{fv}(\sigma_1)\ .$$

Equally as in $\mathbf{HB}_\mathbf{0}^=$, applications of ARROW/FIX are also in $\mathbf{ann\text{-}HB}_\mathbf{0}^=$ subject to the side-condition **I**, which demands that always at least one open marked assumption is actually discharged. For an application of ARROW/FIX as depicted in Figure 8.2 the side-condition **I** demands precisely:

The open assumption class $[\alpha : \tau_1 \to \tau_2 = \sigma_1 \to \sigma_2]^\alpha$ is inhabited in $\mathcal{D}_1$ or in $\mathcal{D}_2$, i.e. there is at least one occurrence of an open assumption $(\alpha : \tau_1 \to \tau_2 = \sigma_1 \to \sigma_2)^\alpha$ in $\mathcal{D}_1$ or in $\mathcal{D}_2$.

⊠

Remark 8.1.2. (a) According to Definition 8.1.1, a possible marked assumption for a derivation in $\mathbf{ann\text{-}HB}_\mathbf{0}^=$ is an equation between recursive types which is annotated by a type variable that does not occur free in the recursive type on either side of the equation and that is also used to mark the entire formula as an assumption. This dual use of a type variable α in a marked assumptions $(\alpha : \tau = \sigma)^\alpha$ both as an annotation for the equation $\tau = \sigma$ and as an assumption marker is in fact redundant: due to the side-condition $\alpha \notin \mathrm{fv}(\tau) \cup \mathrm{fv}(\sigma)$ on a marked assumption $(\alpha : \tau = \sigma)^\alpha$, a formula in a marked assumption cannot be an axiom of $\mathbf{ann\text{-}HB}_\mathbf{0}^=$. This entails that, if assumptions in $\mathbf{ann\text{-}HB}_\mathbf{0}^=$-derivations were not marked, we could still tell from a formula at a leaf-position of an $\mathbf{ann\text{-}HB}_\mathbf{0}^=$-derivation whether it is an axiom or an assumption.

Notwithstanding the possible formal simplification of the system $\mathbf{ann\text{-}HB}_\mathbf{0}^=$ suggested by this observation, assumption markers have been kept here in accordance with the usual notation for derivations in natural-deduction systems.

(b) In Definition 6.2.1 (in Section 6.2 of Chapter 6) we have introduced a general 'circular' rule FIX and shown that it is cr-admissible with respect to the system $\mathbf{HB}_\mathbf{0}^=$. In Remark 6.2.2, (b), we noted that this rule does not fit into the format of ANDS-rule as introduced in Section B.2 of Appendix B. In the context of the system $\mathbf{ann\text{-}HB}_\mathbf{0}^=$ defined above, however, it is possible to introduce a version of the rule FIX that can be formalized as an ANDS-rule.

This is because, for derivations $\mathcal{D}$ with conclusion $\chi : \tau = \sigma$ in $\mathbf{ann\text{-}HB}_\mathbf{0}^=$, a side-condition analogous to (6.4) on the contractiveness (in the sense of Definition 6.2.1) of $\mathcal{D}$ with respect to certain open assumptions $(\alpha : \tau = \sigma)^\alpha$ can be expressed as the condition of contractiveness (in the sense of Definition 5.1.1) of χ with respect to the type variable α. More precisely, the rule FIX can here be defined as the rule with applications of the form

$$\dfrac{\begin{array}{c}[\alpha : \tau = \sigma]^\alpha \\ \mathcal{D}_1 \\ \chi : \tau = \sigma\end{array}}{\mu\alpha.\,\chi : \tau = \sigma}\,\mathrm{FIX}\quad(\text{if } \alpha \downarrow \chi) \tag{8.1}$$

(with arbitrary $\alpha \in TVar$ and $\tau, \sigma, \chi \in \mu Tp$). In applications of the form (8.1) no explicit restriction is imposed on the form of the subderivation $\mathcal{D}_1$; however, it is easy to prove that the side-condition $\alpha \downarrow \chi$ guarantees that the following assertion holds about the structure of $\mathcal{D}_1$:

"There are *either no* undischarged marked assumptions in $\mathcal{D}_1$ of the form $(\alpha : \tau = \sigma)^{\alpha}$ *or* there is *at least one* occurrence of an application of ARROW or ARROW/FIX in $\mathcal{D}_1$."

Now is easy to see that, unlike the rule FIX defined in Definition 6.2.1, the rule FIX with applications defined according to (8.1) *can* in fact *be formalized* as an ANDS-rule.

We mention here without proof the fact that the rule FIX with applications of the form (8.1) is cr-admissible in **ann-HB**$_{\mathbf{0}}^{=}$ and that furthermore every derivation $\mathcal{D}$ in **ann-HB**$_{\mathbf{0}}^{=}$+FIX can effectively be transformed into a mimicking derivation $\mathcal{D}'$ in **ann-HB**$_{\mathbf{0}}^{=}$; this can be shown analogously as in the proof of Lemma 6.2.3. What is more, an annotated version **ann-e-HB**$_{\mathbf{0}}^{=}$ of the system **e-HB**$_{\mathbf{0}}^{=}$ from Definition 6.2.4 can be given (by introducing annotated versions of the rules REN/FIX, FOLD$_l$/FIX, and FOLD$_r$/FIX in a similar way as an annotated version of FIX is defined according to (8.1)) and shown to be equivalent to **ann-HB**$_{\mathbf{0}}^{=}$; this can be demonstrated by a proof analogous to the one given for Theorem 6.2.6.

For the formulation of statements, and for proofs in this section, we will use the following convention for simple symbolic prooftrees with open assumptions in **HB**$^{=}$, **HB**$_{\mathbf{0}}^{=}$ and in **ann-HB**$_{\mathbf{0}}^{=}$.

Notation 8.1.3. Let $\mathcal{S}$ be one of the systems **HB**$^{=}$ or **HB**$_{\mathbf{0}}^{=}$. Throughout this subsection an $\mathcal{S}$-derivation $\mathcal{D}$ denoted by a symbolic prooftree of the simple form

$$\begin{array}{c} \{\, [\,\tau_i = \sigma_i\,]^{u_i} \,\}_{i=1,\ldots,n} \\ \mathcal{D} \\ \tau = \sigma \end{array} \tag{8.2}$$

(with some $n \in \omega$, $\tau, \sigma, \tau_1, \ldots, \tau_n, \sigma_1, \ldots, \sigma_n \in \mu Tp$ and some assumption markers $u_1, \ldots, u_n$) that is considered independently and not as an occurrence within a more complicated prooftree will be understood in the following way: as a derivation $\mathcal{D}$ in the system $\mathcal{S}$ with conclusion $\tau = \sigma$, whose open and inhabited assumption classes are precisely those that are members of the family $\{\, [\,\tau_i = \sigma_i\,]^{u_i} \,\}_{i=1,\ldots,n}$ of assumption classes which is displayed in (8.2) at the top. Spelt out in more detail, this means: if $\mathcal{D}$ contains open assumptions, then $n = 0$ is the case, otherwise $n \in \omega \backslash \{0\}$ holds. If $n > 0$ holds, then for all open marked assumptions of the form $(\rho_1 = \rho_2)^u$ there must exists $i \in \{1, \ldots, n\}$ such that $u \equiv u_i$, $\rho_1 \equiv \tau_i$ and $\rho_2 \equiv \sigma_i$ are the case; and furthermore, for all $i \in \{1, \ldots, n\}$, there does exist at least one open marked assumption of the form $(\tau_i = \sigma_i)^{u_i}$ in $\mathcal{D}$.

However, this convention will not apply to symbolic prooftrees of more complicated forms, for example, for an $\mathcal{S}$-derivation denoted by a symbolic prooftree

$$\dfrac{\begin{array}{c}\big\{[\tau_i = \sigma_i]^{u_i}\big\}_{i=1,\ldots,n}\\ \mathcal{D}_1\\ \tilde{\tau}_1 = \tilde{\sigma}_1\end{array}\qquad\qquad \begin{array}{c}\big\{[\tau_i = \sigma_i]^{u_i}\big\}_{i=1,\ldots,n}\\ \mathcal{D}_2\\ \tilde{\tau}_2 = \tilde{\sigma}_2\end{array}}{\tilde{\tau}_1 \to \tilde{\tau}_2 = \tilde{\sigma}_1 \to \tilde{\sigma}_2}\,\text{ARROW} \tag{8.3}$$

(with some $n \in \omega$, $\tau, \sigma, \tau_1, \ldots, \tau_n, \tilde{\tau}_1, \tilde{\tau}_2, \sigma_1, \ldots, \sigma_n, \tilde{\sigma}_1, \tilde{\sigma}_2 \in \mu Tp$ and assumption markers $u_1, \ldots, u_n$) ending with an application of ARROW. In particular, it does not apply to the symbolic prooftrees denoting the immediate subderivations $\mathcal{D}_1$ and $\mathcal{D}_2$ of $\mathcal{D}$ within the symbolic prooftree (8.3) for $\mathcal{D}$: we do not assume that every assumption $[\tau_i = \sigma_i]^{u_i}$, for $(\tau_i = \sigma_i)^{u_i}$, does in fact occur as an open assumption in both the subderivations $\mathcal{D}_1$ and $\mathcal{D}_2$. In this case we will generally only assume that every assumption $[\tau_i = \sigma_i]^{u_i}$, for $i \in \{1, \ldots, n\}$, does in fact occur as an open assumption in one of $\mathcal{D}_1$ or $\mathcal{D}_2$; however this will be spelt out explicitly in such situations.

The use of this notation will also be extended analogously to the symbolic denotation of derivations in the annotated system **ann-HB**$_{\mathbf{0}}^{=}$.

⊠

The side-condition **S** on the annotations in open marked assumptions of the immediate subderivations of an **ann-HB**$_{\mathbf{0}}^{=}$-derivation that ends with an application of ARROW or ARROW/FIX implies very similar, but in fact stronger statements than the conditions (a), (b), and (c) in Definition 8.1.1. These stronger statements are the respective assertions (a), (b), and (c) in the following lemma, which we will need later.

Lemma 8.1.4. *Let $\mathcal{D}$ be a derivation in* **ann-HB**$_{\mathbf{0}}^{=}$ *with conclusion $\chi : \tau = \sigma$, where $\chi, \tau, \sigma \in \mu Tp$.*

Suppose that, for some $n \in \omega$, recursive types $\tau, \sigma, \chi, \tau_1, \ldots, \tau_n, \sigma_1, \ldots, \sigma_n$, and type variables $\alpha_1, \ldots, \alpha_n$, the derivation $\mathcal{D}$ is of the form

$$\begin{array}{c}\{\,[\,\alpha_i \,:\, \tau_i = \sigma_i\,]^{\alpha_i}\,\}_{i=1,\ldots,n}\\ \mathcal{D}\\ \chi \,:\, \tau = \sigma\end{array} \tag{8.4}$$

and that furthermore, for some $m \in \omega$, $\{\bar{\tau}_1, \ldots, \bar{\tau}_m\}$ is the set of all recursive types $\bar{\tau}$ that appear in occurrences of axioms (REFL) *$\bar{\tau} : \bar{\tau} = \bar{\tau}$ at the top of $\mathcal{D}$.*

Then the following three statements hold about the open marked assumptions of $\mathcal{D}$:

(a) In different open assumption classes of $\mathcal{D}$ different type variables are used as annotations. With the denotations used here for $\mathcal{D}$, this means:

$$(\forall i \in \{1, \ldots, n\})\,(\,i \neq j \;\Rightarrow\; \alpha_i \not\equiv \alpha_j\,)\,.$$

(b) *Type variables used as annotations in open marked assumptions of $\mathcal{D}$ do not have free occurrences in the recursive types appearing in axioms* (REFL) *situated at the top of the prooftree $\mathcal{D}$. Here this means:*

$$(\,\forall i \in \{1,\dots,n\}\,)\,(\,\forall j \in \{1,\dots,m\}\,)\,(\,\alpha_i \notin fv(\bar{\tau}_j)\,)\;.$$

(c) *Type variables that are used as annotations in open marked assumptions in $\mathcal{D}$ do not have free occurrences in recursive types that appear on either side of the equation in an open marked assumption in $\mathcal{D}$. More precisely,*

$$(\,\forall i,j \in \{1,\dots,n\}\,)\,(\,\alpha_i \notin fv(\tau_j) \;\wedge\; \alpha_i \notin fv(\sigma_j)\,)\;.$$

must hold with respect to the denotations used here.

Furthermore it holds that

$$fv(\chi) = \;\{\alpha_1,\dots,\alpha_n\} \;\uplus\; \bigcup_{i=1}^{m} fv(\bar{\tau}_i)\;\;, \tag{8.5}$$

where the symbol $\uplus$ is used to designate a disjoint union of sets.

Proof. The lemma can be shown by induction on the depth $|\mathcal{D}|$ of a derivation $\mathcal{D}$ in **ann-HB$_{\mathbf{0}}^{=}$** of the form (8.4).

For the base case of the induction, we have to consider a derivation $\mathcal{D}$ in **ann-HB$_{\mathbf{0}}^{=}$** of the form (8.4) with $|\mathcal{D}| = 0$. This entails that $\mathcal{D}$ is either an axiom or a marked assumption. If $\mathcal{D}$ is an axiom (REFL) of the form $\tau : \tau = \tau$, then the statements (a), (b) and (c) are empty conditions, and (8.5) is the assertion $\mathrm{fv}(\tau) = \emptyset \uplus \mathrm{fv}(\tau)$, which holds obviously. If, on the other hand, $\mathcal{D}$ is a marked assumption $(\alpha : \tau = \sigma)^{\alpha}$, then statement (a) is trivial, statement (b) is an empty condition, and statement (c) consists of the parts $\alpha \notin \mathrm{fv}(\tau)$ and $\alpha \notin \mathrm{fv}(\sigma)$, both of which are true assertions due to the side-condition that must be fulfilled by marked assumptions appearing in **ann-HB$_{\mathbf{0}}^{=}$**-derivations.

For the treatment of the induction step, let now $\mathcal{D}$ be an arbitrary derivation in **ann-HB$_{\mathbf{0}}^{=}$** of the form (8.4) with $|\mathcal{D}| > 0$. If the last rule application in $\mathcal{D}$ is that of a one-premise rule, then the statements to show coincide with respective parts of the induction hypothesis; this is because one-premise rules do neither discharge assumptions nor change the annotation of the equation in their premise. Hence in this case there remains nothing to be shown for the induction hypothesis.

The remaining cases, in which the last rule application in $\mathcal{D}$ is either that of an ARROW- or that of an ARROW/FIX-rule, are very similar to treat. Here we will consider in some detail only the case with an ARROW-rule, in which $\mathcal{D}$ can be written as

$$\begin{array}{c}
\begin{array}{cc}
\big\{[\alpha'_{i_1} : \tau'_{i_1} = \sigma'_{i_1}]^{\alpha'_{i_1}}\big\}_{i_1} & \big\{[\alpha''_{i_2} : \tau''_{i_2} = \sigma''_{i_2}]^{\alpha''_{i_2}}\big\}_{i_2} \\
\big\{[\alpha'''_{i_3} : \tau'''_{i_3} = \sigma'''_{i_3}]^{\alpha'''_{i_3}}\big\}_{i_3} & \big\{[\alpha'''_{i_3} : \tau'''_{i_3} = \sigma'''_{i_3}]^{\alpha'''_{i_3}}\big\}_{i_3} \\
\mathcal{D}_1 & \mathcal{D}_2 \\
\chi_1 : \tilde{\tau}_1 = \tilde{\sigma}_1 & \chi_2 : \tilde{\tau}_2 = \tilde{\sigma}_2
\end{array} \\ \hline
\chi_1 \to \chi_2 : \tilde{\tau}_1 \to \tilde{\tau}_2 = \tilde{\sigma}_1 \to \tilde{\sigma}_2
\end{array}\;\mathrm{ARROW} \tag{8.6}$$

where, for some $n_1, n_2, n_3 \in \omega$, the indices i_1, i_2 and i_3 range, respectively, over the sets $\{1,\ldots,n_1\}$, $\{1,\ldots,n_2\}$, and $\{1,\ldots,n_3\}$; furthermore $\big\{[\alpha'_{i_1} : \tau'_{i_1} = \sigma'_{i_1}]^{\alpha'_{i_1}}\big\}_{i_1}$ denotes the family of those open assumption classes in $\mathcal{D}_2$ that are inhabited only in $\mathcal{D}_1$ (but not in $\mathcal{D}_2$), accordingly $\big\{[\alpha''_{i_2} : \tau''_{i_2} = \sigma''_{i_2}]^{\alpha''_{i_2}}\big\}_{i_2}$ denotes the family of those open assumption classes in $\mathcal{D}_2$ that are inhabited only in $\mathcal{D}_2$ (but not in $\mathcal{D}_1$), and the respective families $\{[\alpha'''_{i_3} : \tau'''_{i_3} = \sigma'''_{i_3}]^{\alpha'''_{i_3}}\}_{i_3}$ indicated at the top of $\mathcal{D}_1$ and $\mathcal{D}_2$ denote the family of the respective parts in $\mathcal{D}_1$ and $\mathcal{D}_2$ of all those open assumption classes of $\mathcal{D}$ that are inhabited in both $\mathcal{D}_1$ and $\mathcal{D}_2$.

The conditions (a), (b) and (c) for $\mathcal{D}$ are very similar to verify. They all follow from the induction hypothesis (concerning the assertions (a), (b) and (c)) on $\mathcal{D}_1$ and $\mathcal{D}_2$ together with appeals to the side-condition **S** on the application of ARROW at the bottom of $\mathcal{D}$. Here comes just a little piece of the argument as an example. As part of verifying the condition (c) for $\mathcal{D}$ it has to be checked that α''_j (with $j \in \{1,\ldots,n_2\}$) does not occur in $\mathrm{fv}(\tau''_{i_2}) \cup \mathrm{fv}(\sigma''_{i_2})$ for all $i_2 \in \{1,\ldots,n_2\}$, neither in $\mathrm{fv}(\tau'''_{i_3}) \cup \mathrm{fv}(\sigma'''_{i_3})$ for all $i_3 \in \{1,\ldots,n_3\}$, nor in $\mathrm{fv}(\tau'_{i_1}) \cup \mathrm{fv}(\sigma'_{i_1})$ for all $i_1 \in \{1,\ldots,n_1\}$. Clearly the first two assertions follow from the induction hypothesis on $\mathcal{D}_2$ (concerning the condition (c)) and the third one follows from the part (c) of the side-condition **S** on the bottommost application of ARROW in $\mathcal{D}$.

For showing the assertion (8.5) for $\mathcal{D}$, let the sets $\{\bar\tau_1,\ldots,\bar\tau_m\}$, $\{\bar\tau'_1,\ldots,\bar\tau'_{m_1}\}$, and $\{\bar\tau''_1,\ldots,\bar\tau''_{m_2}\}$, where $m, m_1, m_2 \in \omega$, be the sets of all recursive types $\bar\tau$ that occur in axioms $\bar\tau : \bar\tau = \bar\tau$ at the top of $\mathcal{D}$, $\mathcal{D}_1$, and $\mathcal{D}_2$, respectively. Hence it is clear that

$$\{\bar\tau_1,\ldots,\bar\tau_m\} = \{\bar\tau'_1,\ldots,\bar\tau'_{m_1},\bar\tau''_1,\ldots,\bar\tau''_{m_2}\}\ . \tag{8.7}$$

Since $\mathcal{D}$ was assumed to be both of the form (8.4) and of the form (8.6), it follows that

$$\{\alpha_1,\ldots,\alpha_n\} = \{\alpha'_1,\ldots,\alpha'_{n_1},\alpha''_1,\ldots,\alpha''_{n_2},\alpha'''_1,\ldots,\alpha'''_{n_3}\}\ . \tag{8.8}$$

Two applications of the induction hypothesis (more precisely, that part of it which concerns the assertion (8.5)) to $\mathcal{D}_1$ and to $\mathcal{D}_2$ give the statements

$$\left.\begin{aligned} \mathrm{fv}(\chi_1) &= \{\alpha'_1,\ldots,\alpha'_{n_1},\alpha'''_1,\ldots,\alpha'''_{n_3}\} \uplus \bigcup_{j=1}^{m_1} \mathrm{fv}(\bar\tau'_j)\ , \\ \mathrm{fv}(\chi_2) &= \{\alpha''_1,\ldots,\alpha''_{n_2},\alpha'''_1,\ldots,\alpha'''_{n_3}\} \uplus \bigcup_{j=1}^{m_2} \mathrm{fv}(\bar\tau''_j)\ . \end{aligned}\right. \tag{8.9}$$

Since by the part (b) of the assertion of the induction step, the proof of which part we have hinted at above, no type variable used as an annotation in either $\mathcal{D}_1$ and $\mathcal{D}_2$ does occur free in an axiom at the top of $\mathcal{D}$, it follows from (8.7), (8.8) and (8.9):

$$\begin{aligned} \mathrm{fv}(\chi_1 \to \chi_2) &= \{\ldots,\alpha'_{i_1},\ldots,\alpha''_{i_2},\ldots,\alpha'''_{i_3},\ldots\} \uplus \Big(\bigcup_{j=1}^{m_1} \mathrm{fv}(\bar\tau'_j) \cup \bigcup_{j=1}^{m_2} \mathrm{fv}(\bar\tau''_j)\Big) \\ &= \{\alpha_1,\ldots,\alpha_n\} \uplus \bigcup_{j=1}^{m} \mathrm{fv}(\bar\tau_j)\ . \end{aligned}$$

This shows the part concerning the assertion (8.5) of the induction step in the case considered here of an application of an ARROW-rule at the bottom of $\mathcal{D}$.

The induction step for the case of an application of ARROW/FIX at the bottom of $\mathcal{D}$ can be treated very similarly, and what concerns the part regarding the assertion (8.5), with just slightly more effort. It is clear that, relative to similar denotations as the ones used above for the case with an ARROW-rule, the annotation α_0 of the open assumption class $[\alpha_0 : \tilde{\tau}_1 \to \tilde{\tau}_2 = \tilde{\sigma}_1 \to \tilde{\sigma}_2]^{\alpha_0}$, which is discharged by a considered application of ARROW/FIX, disappears from $\mathrm{fv}(\mu\alpha_0.\,(\chi_1 \to \chi_2))$ for the annotation in the conclusion

$$\mu\alpha_0.\,(\chi_1 \to \chi_2) : \tilde{\tau}_1 \to \tilde{\tau}_2 = \tilde{\sigma}_1 \to \tilde{\sigma}_2$$

of $\mathcal{D}$. But it has to be checked that no other variable besides α_0 disappears from the set $\mathrm{fv}(\chi_1 \to \chi_2) = \mathrm{fv}(\chi_1) \cup \mathrm{fv}(\chi_2)$ (due to side-condition **I** on the considered ARROW/FIX-application and the induction hypothesis we indeed find $\alpha_0 \in \mathrm{fv}(\chi_1 \to \chi_2)$); this ultimately follows from the fulfilledness of the conditions (a), (b) and (c) for $\mathcal{D}$ as can be argued analogously to the case treated above of an application of ARROW at the bottom of a derivation of the form (8.4).

□

The following lemma concerns the relationship between derivations in the system $\mathbf{HB}_{\mathbf{0}}^{=}$ and derivations in the annotated system $\textbf{ann-}\mathbf{HB}_{\mathbf{0}}^{=}$. By its proof a transformation from derivations in $\mathbf{HB}_{\mathbf{0}}^{=}$ to derivations in $\textbf{ann-}\mathbf{HB}_{\mathbf{0}}^{=}$ is defined that proceeds by finding appropriate annotations, and another one in the opposite direction that proceeds by 'forgetting' annotations. The *annotation step* of our transformation between $\mathbf{HB}_{\mathbf{0}}^{=}$ and $\mathbf{AC}^{=}$ is guaranteed by the statement in item (i) of the lemma and by the mentioned effective transformation of $\mathbf{HB}_{\mathbf{0}}^{=}$-derivations into $\textbf{ann-}\mathbf{HB}_{\mathbf{0}}^{=}$-derivations.

Lemma 8.1.5. *The following two kinds of effective transformations are possible between derivations in the system* $\mathbf{HB}_{\mathbf{0}}^{=}$ *and derivations in the annotated system* $\textbf{ann-}\mathbf{HB}_{\mathbf{0}}^{=}$*:*

(i) Every derivation $\mathcal{D}$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ *can effectively be annotated by appropriate recursive types with a derivation $\hat{\mathcal{D}}$ in* $\textbf{ann-}\mathbf{HB}_{\mathbf{0}}^{=}$ *as the result. More formally, the statement (i) of the Precise Formulation, which is given below, holds.*

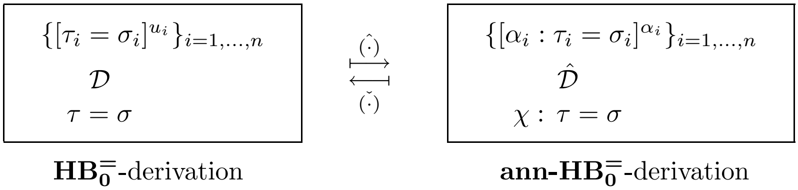
(ii) Conversely, every derivation $\mathcal{D}$ in $\textbf{ann-}\mathbf{HB}_{\mathbf{0}}^{=}$ *can effectively be transformed into a* $\mathbf{HB}_{\mathbf{0}}^{=}$*-derivation $\check{\mathcal{D}}$ by stripping it of its annotations. More precisely, the assertion (ii) of the Precise Formulation, which is stated below, holds.*

Precise Formulation. The statements (i) and (ii) of the lemma refer to the following two refined statements (i) and (ii), respectively, which are illustrated together by Figure 8.3.

(i) Let an arbitrary derivation $\mathcal{D}$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ of the form

$$\begin{array}{c} \{\, [\,\tau_i = \sigma_i\,]^{u_i}\,\}_{i=1,\ldots,n} \\ \mathcal{D} \\ \tau = \sigma \end{array} \qquad (8.10)$$

Figure 8.3: Illustration of the statement of Lemma 8.1.5: there exist an annotation transformation $\hat{(\cdot)}$ from derivations in $\mathbf{HB}_{\mathbf{0}}^{=}$ into derivations in $\textbf{ann-HB}_{\mathbf{0}}^{=}$, and an annotation-removing transformation $\check{(\cdot)}$ from derivations in $\textbf{ann-HB}_{\mathbf{0}}^{=}$ into derivations in $\mathbf{HB}_{\mathbf{0}}^{=}$.


$\{[\tau_i = \sigma_i]^{u_i}\}_{i=1,\dots,n}$
$\mathcal{D}$
$\tau = \sigma$
$\hat{(\cdot)}$
$\check{(\cdot)}$
$\{[\alpha_i : \tau_i = \sigma_i]^{\alpha_i}\}_{i=1,\dots,n}$
$\hat{\mathcal{D}}$
$\chi : \tau = \sigma$
$\mathbf{HB}_{\mathbf{0}}^{=}$-derivation
$\textbf{ann-HB}_{\mathbf{0}}^{=}$-derivation

with $n \in \omega$ and recursive types τ, σ and τ_i, σ_i for all $i \in \{1, \dots, n\}$ be given. Then there exist annotations consisting of recursive types for each formula, i.e. equation between recursive types, in $\mathcal{D}$, an annotation $\chi \in \mu Tp$ for the conclusion $\tau = \sigma$ of $\mathcal{D}$ and different type variables $\alpha_1, \dots, \alpha_n$ as annotations for the open assumption classes of $\mathcal{D}$, such that the result of prefixing the respective annotations to all formulas occurring in $\mathcal{D}$ is an $\textbf{ann-HB}_{\mathbf{0}}^{=}$-derivation of the form

$$\begin{array}{c} \{\,[\,\alpha_i \,:\, \tau_i = \,\sigma_i\,]^{\alpha_i}\,\}_{i=1,\dots,n} \\ \hat{\mathcal{D}} \\ \chi \,:\, \tau = \,\sigma \end{array} \tag{8.11}$$

(ii) Let $\mathcal{D}$ be a derivation in $\textbf{ann-HB}_{\mathbf{0}}^{=}$ of the form

$$\begin{array}{c} \{\,[\,\alpha_i \,:\, \tau_i = \,\sigma_i\,]^{\alpha_i}\,\}_{i=1,\dots,n} \\ \mathcal{D} \\ \chi \,:\, \tau = \,\sigma \end{array}$$

with some $n \in \omega$, recursive types τ, σ and τ_i, σ_i for all $i = 1, \dots, n$ as well as with type variables $\alpha_1, \dots, \alpha_n$. Then the result $\check{\mathcal{D}}$ of replacing, for all $\tilde{\tau}, \tilde{\sigma}, \tilde{\chi} \in \mu Tp$, each occurrence of an $\textbf{ann-HB}_{\mathbf{0}}^{=}$-formula $\tilde{\chi} : \tilde{\tau} = \tilde{\sigma}$ in the prooftree $\mathcal{D}$ by an occurrence of the $\mathbf{HB}_{\mathbf{0}}^{=}$-formula $\tilde{\tau} = \tilde{\sigma}$, thereby also replacing each occurrences of a marked assumption $(\tilde{\alpha} : \tilde{\tau} = \tilde{\sigma})^{\tilde{\alpha}}$ (for some $\tilde{\alpha} \in TVar$) by an occurrences of a marked assumption $(\tilde{\tau} = \tilde{\sigma})^{\tilde{\alpha}}$, is a derivation in $\mathbf{HB}_{\mathbf{0}}^{=}$ of the form

$$\begin{array}{c} \{\,[\,\tau_i = \,\sigma_i\,]^{\alpha_i}\,\}_{i=1,\dots,n} \\ \check{\mathcal{D}} \\ \tau = \,\sigma \end{array}$$

(hereby the type variables used as annotations and assumption markers in marked assumptions of $\mathcal{D}$ are retained as assumption markers for respective corresponding assumptions in $\check{\mathcal{D}}$).

Proof. The parts (i) and (ii) of the lemma are respectively treated in the items (a) and (b) below.

(a) It suffices to show the statement (i) in the Precise Formulation of the lemma *only for such derivations* of the form (8.10), in which the assumption markers are *type variables* with the property that (1) type variables used as markers for different assumption classes (whether discharged or open) in $\mathcal{D}$ are different, and that (2) type variables used as assumption markers in $\mathcal{D}$ do not occur in recursive types $\tilde{\tau}$ or $\tilde{\sigma}$ for any equation between recursive types $\tilde{\tau} = \tilde{\sigma}$ in $\mathcal{D}$.

This simplification in what has to be shown is justified because: it is clearly possible to replace the assumption markers in a derivation of the form (8.10) by type variables that did not previously occur in the derivation. This can be done in such a way that different type variables are used to annotate different assumption classes (whether ultimately discharged or undischarged ones) in $\mathcal{D}$. Then clearly the above mentioned properties (1) and (2) are satisfied for the annotations of marked assumptions in the resulting derivation.

But now the definition of a transformation of a derivation $\mathcal{D}$ in $\mathbf{HB}^{=}$ of the form (8.10), where the type variables used as assumption markers fulfill the above conditions (1) and (2), into a derivation $\hat{\mathcal{D}}$ in $\mathbf{ann\text{-}HB}_{\mathbf{0}}^{=}$ of the form (8.11) is a matter of a straightforward induction. Because many of the involved finer details occur in a very similar way in the proofs of Lemma 8.1.4 and of Lemma 8.1.6, some of them will not be explained in full precision here.

For an outline of this inductive definition, let $\mathcal{D}$ be an arbitrary derivation in $\mathbf{HB}_{\mathbf{0}}^{=}$ of the form (8.10), where the type variables used as assumption markers fulfill the conditions (1) and (2) above.

In the base case of the induction, where $|\mathcal{D}| = 0$, $\mathcal{D}$ is either an axiom (REFL) of $\mathbf{HB}_{\mathbf{0}}^{=}$ of the form $\tau = \tau$ with some $\tau \in \mu Tp$ or a marked assumption of the form $(\tau = \sigma)^{\alpha}$ with $\tau, \sigma \in \mu Tp$ and $\alpha \in TVar$ such that $\alpha \notin \mathrm{fv}(\tau) \cup \mathrm{fv}(\sigma)$ (since the condition (2) is satisfied for $\mathcal{D}$). In the first case $\hat{\mathcal{D}}$ can obviously be defined as $\tau : \tau = \tau$ and in the second case as the marked assumption $(\alpha : \tau = \sigma)^{\alpha}$ (which is allowed in $\mathbf{ann\text{-}HB}_{\mathbf{0}}^{=}$ because of $\alpha \notin \mathrm{fv}(\tau) \cup \mathrm{fv}(\sigma)$).

For the induction step, we assume that the derivation $\mathcal{D}$ is as above and has depth $|\mathcal{D}| \geq 1$. We distinguish the cases, in which the last rule application in $\mathcal{D}$ is that of a one-premise rule or that of a two-premise rule of $\mathbf{ann\text{-}HB}_{\mathbf{0}}^{=}$.

In the case of a one-premise rule, the derivation $\mathcal{D}$ is of the form

$$\begin{array}{c} \{\, [\,\tau_i = \sigma_i\,]^{\alpha_i}\,\}_{i=1,\ldots,n} \\ \mathcal{D}_1 \\ \dfrac{\tilde{\tau} = \tilde{\sigma}}{\tau = \sigma}\, R \end{array}$$

with some recursive types $\tilde{\tau}, \tilde{\sigma}$ and τ_i, σ_i for all $i \in \{1, \ldots, n\}$ and type variables $\alpha_1, \ldots, \alpha_n$, such that furthermore the conditions (1) and (2) are fulfilled, and where R is one of the rules $\mathrm{FOLD}_{l/r}$ or $(\mu\text{–}\bot)_{l/r}^{\bot\mathrm{der}}$. Due to the induction hypothesis it is possible to produce effectively, starting from the subderivation $\mathcal{D}_1$ of $\mathcal{D}$, a derivation $\hat{\mathcal{D}}_1$ in $\mathbf{ann\text{-}HB}_{\mathbf{0}}^{=}$ with the conclusion $\chi : \tilde{\tau} = \tilde{\sigma}$ for some annotation $\chi \in \mu Tp$ and with the open assumption classes $[\alpha_i : \tau_i = \sigma_i]^{\alpha_i}$ for $i \in \{1, \ldots, n\}$. Choosing $\hat{\mathcal{D}}_1$ in this way the desired derivation $\hat{\mathcal{D}}$ can then be defined as of the form

$$\dfrac{\begin{array}{c}\{\,[\,\alpha_i\,:\,\tau_i = \,\sigma_i\,]^{\alpha_i}\,\}_{i=1,\ldots,n}\\ \hat{\mathcal{D}}_1\\ \chi\,:\,\tilde{\tau} = \,\tilde{\sigma}\end{array}}{\chi\,:\,\tau = \,\sigma}\,R$$

in $\mathbf{ann\text{-}HB}_{\mathbf{0}}^{=}$, where the rule application labeled by R at the bottom of $\hat{\mathcal{D}}$ is an application of a rule in $\mathbf{ann\text{-}HB}_{\mathbf{0}}^{=}$ *of the same kind* as the application labeled by R stood for as last rule application in $\mathcal{D}$. This definition is indeed justified, since in all cases of one-premise rules of $\mathbf{ann\text{-}HB}_{\mathbf{0}}^{=}$ the annotation in the premise does not differ from the annotation in the conclusion.

In the case of a two-premise rule, the bottommost rule application in $\mathcal{D}$ is one of the rule ARROW or one of the rule ARROW/FIX. We consider only the definition of $\hat{\mathcal{D}}$ from $\mathcal{D}$ the subcase with an application of ARROW/FIX because the case with an application of ARROW can be settled analogously and easier. In this subcase the $\mathbf{HB}_{\mathbf{0}}^{=}$-derivation $\mathcal{D}$ can be written as of the form

$$\dfrac{\begin{array}{c}[\tilde{\tau}_1 \to \tilde{\tau}_2 = \tilde{\sigma}_1 \to \tilde{\sigma}_2]^{\alpha_0}\\ \left\{[\tau_i = \sigma_i]^{\alpha_i}\right\}_{i=1,\ldots,n}\\ \mathcal{D}_1\\ \tilde{\tau}_1 = \tilde{\sigma}_1\end{array} \qquad \begin{array}{c}[\tilde{\tau}_1 \to \tilde{\tau}_2 = \tilde{\sigma}_1 \to \tilde{\sigma}_2]^{\alpha_0}\\ \left\{[\tau_i = \sigma_i]^{\alpha_i}\right\}_{i=1,\ldots,n}\\ \mathcal{D}_2\\ \tilde{\tau}_2 = \tilde{\sigma}_2\end{array}}{\tilde{\tau}_1 \to \tilde{\tau}_2 = \tilde{\sigma}_1 \to \tilde{\sigma}_2}\,\text{ARROW/FIX}, \alpha_0$$

for some $\tilde{\tau}_1, \tilde{\tau}_2, \tilde{\sigma}_1, \tilde{\sigma}_2$ and a type variable α_0, where due to the side-condition **I** on the application of ARROW/FIX at the bottom of $\mathcal{D}$, the assumption class $[\tilde{\tau}_1 \to \tilde{\tau}_2 = \tilde{\sigma}_1 \to \tilde{\sigma}_2]^{\alpha_0}$ is inhabited in at least one of the immediate subderivations $\mathcal{D}_1$ and $\mathcal{D}_2$ of $\mathcal{D}$. By the induction hypothesis the two $\mathbf{ann\text{-}HB}_{\mathbf{0}}^{=}$-derivations $\hat{\mathcal{D}}_1$ and $\hat{\mathcal{D}}_2$ can be generated effectively that have the respective conclusions $\chi_1 : \tilde{\tau}_1 = \tilde{\sigma}_1$ and $\chi_2 : \tilde{\tau}_2 = \tilde{\sigma}_2$ for some $\chi_1, \chi_2 \in \mu Tp$ and that have open assumption classes *among* $[\alpha_i : \tau_i = \sigma_i]^{\alpha_i}$ for $i \in \{2, \ldots, n\}$ and $[\alpha_0 : \tilde{\tau}_1 \to \tilde{\tau}_2 = \tilde{\sigma}_1 \to \tilde{\sigma}_2]^{\alpha_0}$. We choose $\hat{\mathcal{D}}_1$ and $\hat{\mathcal{D}}_2$ in this way and observe that the induction hypothesis also entails that the aforementioned $n + 1$ assumption classes are precisely those open assumption classes that are inhabited in at least one of $\mathcal{D}_1$ or $\mathcal{D}_2$. Now $\hat{\mathcal{D}}$ can be defined as the derivation

$$\dfrac{\begin{array}{c}[\alpha_0 : \tilde{\tau}_1 \to \tilde{\tau}_2 = \tilde{\sigma}_1 \to \tilde{\sigma}_2]^{\alpha_0} \\ \{[\alpha_i : \tau_i = \sigma_i]^{\alpha_i}\}_{i=1,\ldots,n} \\ \hat{\mathcal{D}}_1 \\ \chi_1 : \tilde{\tau}_1 = \tilde{\sigma}_1\end{array} \qquad \begin{array}{c}[\alpha_0 : \tilde{\tau}_1 \to \tilde{\tau}_2 = \tilde{\sigma}_1 \to \tilde{\sigma}_2]^{\alpha_0} \\ \{[\alpha_i : \tau_i = \sigma_i]^{\alpha_i}\}_{i=1,\ldots,n} \\ \hat{\mathcal{D}}_2 \\ \chi_2 : \tilde{\tau}_2 = \tilde{\sigma}_2\end{array}}{\mu\alpha_0.(\chi_1 \to \chi_2) : \tilde{\tau}_1 \to \tilde{\tau}_2 = \tilde{\sigma}_1 \to \tilde{\sigma}_2}\ \text{ARROW/FIX},\ \alpha_0$$

in $\mathbf{ann\text{-}HB}_{\mathbf{0}}^{=}$. That the side-condition **S** is satisfied for the application of ARROW/FIX at the bottom of $\hat{\mathcal{D}}$ is a consequence of the conditions (1) and (2) that we assumed for $\mathcal{D}$, i.e. it is a consequence of special way how assumption markers have been chosen for $\mathcal{D}$. The fulfilledness of the side-condition **I** on the application of ARROW/FIX at the bottom of $\hat{\mathcal{D}}$ follows from the side-condition **I** on the application of ARROW/FIX at the bottom of $\mathcal{D}$ together with what the induction hypothesis implies about the assumption classes of $\mathcal{D}_1$ and $\mathcal{D}_2$. And furthermore, similar reasoning shows that $\hat{\mathcal{D}}$ indeed possesses exactly the family $\{[\alpha_i : \tau_i = \sigma_i]^{\alpha_i}\}_{i=1,\ldots,n}$ of open assumption classes.

(b) This can be shown by straightforward induction on the depth $|\mathcal{D}|$ of $\mathcal{D}$.

□

The following lemma justifies the *extraction* and *combination steps*, outlined at the start of this section, of the transformation from $\mathbf{HB}_{\mathbf{0}}^{=}$-derivations into $\mathbf{AC}^{=}$-derivations. It asserts in particular that two $\mathbf{AC}^{=}$-derivations can be 'extracted' from every derivation $\mathcal{D}$ in the annotated variant-Brandt-Henglein system $\mathbf{ann\text{-}HB}_{\mathbf{0}}^{=}$ with conclusion $\chi : \tau = \sigma$, and that these two derivations can be combined into an $\mathbf{AC}^{=}$-derivation with the conclusion $\tau = \sigma$.

Lemma 8.1.6. *The following two statements hold concerning transformations of derivations in* $\mathbf{ann\text{-}HB}_{\mathbf{0}}^{=}$ *into derivations in* $\mathbf{AC}^{=}$*:*

(i) For every derivation $\mathcal{D}$ in $\mathbf{ann\text{-}HB}_{\mathbf{0}}^{=}$ *that is of the form*

$$\begin{array}{c}\{\,[\,\alpha_i\ :\ \tau_i = \ \sigma_i\,]^{\alpha_i}\,\}_{i=1,\ldots,n} \\ \mathcal{D} \\ \chi\ :\ \tau = \ \sigma\end{array} \tag{8.12}$$

with some $n \in \omega$, $\chi, \tau, \sigma, \tau_1, \ldots, \tau_n, \sigma_1, \ldots, \sigma_n \in \mu Tp$, and $\alpha_1, \ldots, \alpha_n \in TVar$, two $\mathbf{AC}^{=}$*-derivations $\mathcal{D}^{(1)}$ and $\mathcal{D}^{(2)}$ without assumptions and of the respective form*

$$\begin{array}{c}\mathcal{D}^{(1)} \\ \tau = \ \chi[\tau_1/\alpha_1, \ldots, \tau_n/\alpha_n]\end{array} \qquad \begin{array}{c}\mathcal{D}^{(2)} \\ \sigma = \ \chi[\sigma_1/\alpha_1, \ldots, \sigma_n/\alpha_n]\end{array} \tag{8.13}$$

can effectively be constructed from $\mathcal{D}$.

(ii) Every derivation $\mathcal{D}$ in $\mathbf{ann\text{-}HB}_{\mathbf{0}}^{=}$ *without open assumption classes and with conclusion $\chi : \tau = \sigma$, for some $\chi, \tau, \sigma \in \mu Tp$, can effectively be transformed into a derivation $\mathcal{D}'$ in* $\mathbf{AC}^{=}$ *without assumptions and with the conclusion $\tau = \sigma$.*

Since, as mentioned earlier, it is slightly more immediate to extract $\mathbf{AC}^{=}_{*-}$-derivations than $\mathbf{AC}^{=}$-derivations from $\mathbf{ann\text{-}HB}^{=}_{\mathbf{0}}$-derivations without open assumptions, we do not prove this lemma directly, but give its *proof* immediately after stating the following lemma (which proof is then but an easy consequence). The lemma below contains the in effect stronger statement of the existence of similar transformations from derivations in $\mathbf{ann\text{-}HB}^{=}_{\mathbf{0}}$ into derivations in the μ-COMPAT-free variant system $\mathbf{AC}^{=}_{*-}$ of the systems $\mathbf{AC}^{=}_{*}$ and $\mathbf{AC}^{=}$. Its proof contains the main, and perhaps single not entirely straightforward, part of the transformation developed in this section.

Lemma 8.1.7. *The following two statements hold concerning transformations of derivations in* $\mathbf{ann\text{-}HB}^{=}_{\mathbf{0}}$ *into derivations in* $\mathbf{AC}^{=}_{*-}$ *:*

(i) For every derivation $\mathcal{D}$ in $\mathbf{ann\text{-}HB}^{=}_{\mathbf{0}}$ *that is of the form* (8.12)*, where $n \in \omega$, $\alpha_1, \ldots, \alpha_n \in TVar$ and $\chi, \tau, \sigma, \tau_1, \ldots, \tau_n, \sigma_1, \ldots, \sigma_n \in \mu Tp$, two* $\mathbf{AC}^{=}_{*-}$*-derivations $\mathcal{D}^{(1)}_{*-}$ and $\mathcal{D}^{(2)}_{*-}$ without assumptions and of the respective form*

$$\begin{array}{c} \mathcal{D}^{(1)}_{*-} \\ \tau = \chi[\tau_1/\alpha_1, \ldots, \tau_n/\alpha_n] \end{array} \qquad\qquad \begin{array}{c} \mathcal{D}^{(2)}_{*-} \\ \sigma = \chi[\sigma_1/\alpha_1, \ldots, \sigma_n/\alpha_n] \end{array} \tag{8.14}$$

can effectively be constructed from $\mathcal{D}$.

(ii) Every derivation $\mathcal{D}$ in $\mathbf{ann\text{-}HB}^{=}_{\mathbf{0}}$ *without open assumption classes and with conclusion $\chi : \tau = \sigma$, for some $\chi, \tau, \sigma \in \mu Tp$, can effectively be transformed into a derivation $\mathcal{D}'_{*-}$ in* $\mathbf{AC}^{=}_{*-}$ *without assumptions and with the conclusion $\tau = \sigma$.*

The *proof* of this lemma is given below, subsequent to its application for obtaining a proof for Lemma 8.1.6.

Proof of Lemma 8.1.6. The items (i) and (ii) of the lemma follow from the respective items (i) and (ii) in Lemma 8.1.7 in view of Corollary 7.1.17. Let us demonstrate this only in the case of item (ii) here. We consider an arbitrary derivation $\mathcal{D}$ in $\mathbf{ann\text{-}HB}^{=}_{\mathbf{0}}$ without open assumptions and with conclusion $\chi : \tau = \sigma$, for some $\chi, \tau, \sigma \in \mu Tp$. By Lemma 8.1.7, (ii), $\mathcal{D}$ can effectively be transformed into a derivation $\mathcal{D}'_{*-}$ in $\mathbf{AC}^{=}_{*-}$ with conclusion $\tau = \sigma$ and without assumptions. And by Corollary 7.1.17 the derivation $\mathcal{D}'_{*-}$ can effectively be transformed into a derivation $\mathcal{D}'$ in $\mathbf{AC}^{=}$ without assumptions and with the same conclusion $\tau = \sigma$. □

Proof of Lemma 8.1.7. (a) We shall first show that item (ii) is an obvious consequence of the statement in item (i) of the lemma.

Suppose that $\mathcal{D}$ is an arbitrary derivation in $\mathbf{ann\text{-}HB}^{=}_{\mathbf{0}}$ without open assumption classes. Let $\chi : \tau = \sigma$ be the conclusion of $\mathcal{D}$, where $\chi, \tau, \sigma \in \mu Tp$.

Since $\mathcal{D}$ does not contain open assumptions, item (i) of the lemma implies that two derivations $\mathcal{D}^{(1)}_{*-}$ and $\mathcal{D}^{(2)}_{*-}$ with the respective conclusions $\tau = \chi$ and $\sigma = \chi$ and without open assumptions can be produced effectively from $\mathcal{D}$; let

$\mathcal{D}^{(1)}_{*-}$ and $\mathcal{D}^{(2)}_{*-}$ be such $\mathbf{AC}^{=}_{\boldsymbol{*}-}$-derivations. Then the derivation $\mathcal{D}'_{*-}$ of the form

$$\dfrac{\begin{array}{c}\mathcal{D}^{(1)}_{*-}\\ \tau=\chi\end{array} \qquad \dfrac{\begin{array}{c}\mathcal{D}^{(2)}_{*-}\\ \sigma=\chi\end{array}}{\chi=\sigma}\,\mathrm{SYMM}}{\tau=\sigma}\,\mathrm{TRANS}$$

is a derivation in $\mathbf{AC}^{=}_{\boldsymbol{*}-}$ without assumptions and with conclusion $\tau=\sigma$; furthermore it can also be built effectively from $\mathcal{D}$.

In this way we have shown that from every $\textbf{ann-}\mathbf{HB}_{\mathbf{0}}^{=}$-derivation $\mathcal{D}$ without open assumption classes and with conclusion $\chi:\ \tau=\sigma$, for some $\chi,\tau,\sigma\in\mu Tp$, a derivation $\mathcal{D}'_{*-}$ in $\mathbf{AC}^{=}_{\boldsymbol{*}-}$ without assumptions and with conclusion $\tau=\sigma$ can be constructed effectively.

(b) Item (i) of the lemma will be shown by induction on the depth $|\mathcal{D}|$ of a derivation $\mathcal{D}$ in $\textbf{ann-}\mathbf{HB}_{\mathbf{0}}^{=}$ of the form (8.12).

For the base case of the induction, let $\mathcal{D}$ be an arbitrary derivation in $\textbf{ann-}\mathbf{HB}_{\mathbf{0}}^{=}$ of the form (8.12) with $|\mathcal{D}|=0$. Then $\mathcal{D}$ is either an axiom of $\textbf{ann-}\mathbf{HB}_{\mathbf{0}}^{=}$ or a marked assumption. In the case that $\mathcal{D}$ is an axiom and hence of the form $\tau:\tau=\tau$ for some $\tau\in\mu Tp$, both $\mathcal{D}^{(1)}_{*-}$ and $\mathcal{D}^{(2)}_{*-}$ can be chosen as the $\mathbf{AC}^{=}_{\boldsymbol{*}-}$-derivation consisting of the axiom $\tau=\tau$. And in the case that $\mathcal{D}$ is a marked assumption of the form $(\alpha:\ \tau=\sigma)^{\alpha}$ for some $\tau,\sigma\in\mu Tp$ and $\alpha\in TVar$, it is clear, because of $\alpha[\tau/\alpha]\equiv\tau$ and $\alpha[\sigma/\alpha]\equiv\sigma$, that the desired $\mathbf{AC}^{=}_{\boldsymbol{*}-}$-derivations $\mathcal{D}^{(1)}_{*-}$ and $\mathcal{D}^{(2)}_{*-}$ can be chosen as the axioms $\tau=\tau$ and $\sigma=\sigma$, respectively.

For the treatment of the induction step, let $\mathcal{D}$ be an arbitrary derivation in $\textbf{ann-}\mathbf{HB}_{\mathbf{0}}^{=}$ of the form (8.12) with $|\mathcal{D}|>0$. We distinguish seven different cases according to which rule of $\textbf{ann-}\mathbf{HB}_{\mathbf{0}}^{=}$ is applied at the bottom of $\mathcal{D}$.

For the five cases, in which the bottommost rule application in $\mathcal{D}$ is that of a one-premise rule of $\textbf{ann-}\mathbf{HB}_{\mathbf{0}}^{=}$, the induction step can be carried out in a very similar way. Here we shall treat the case of the rule FOLD_l as an example: for this purpose we suppose that $\mathcal{D}$ is of the form

$$\dfrac{\begin{array}{c}\{\,[\,\alpha_i\,:\,\tau_i=\sigma_i\,]^{\alpha_i}\,\}_{i=1,\dots,n}\\ \mathcal{D}_1\\ \chi:\ \tilde{\tau}[\mu\tilde{\alpha}.\,\tilde{\tau}/\tilde{\alpha}]=\sigma\end{array}}{\chi:\ \mu\tilde{\alpha}.\,\tilde{\tau}=\sigma}\,\mathrm{FOLD}_l$$

Due to the induction hypothesis, two derivations in $\mathbf{AC}^{=}_{\boldsymbol{*}-}$ of respective form

$$\begin{array}{c}(\mathcal{D}_1)^{(1)}_{*-}\\ \tilde{\tau}[\mu\tilde{\alpha}.\,\tilde{\tau}/\tilde{\alpha}]\ =\ \chi[\tau_1/\alpha_1,\dots,\tau_n/\alpha_n]\end{array}\qquad\qquad \begin{array}{c}(\mathcal{D}_1)^{(2)}_{*-}\\ \sigma\ =\ \chi[\sigma_1/\alpha_1,\dots,\sigma_n/\alpha_n]\end{array}$$

exist and can effectively be constructed from $\mathcal{D}_1$. Now for the $\mathbf{AC}^{=}_{*-}$-derivation $\mathcal{D}^{(2)}_{*-}$ of the respective form in (8.13), which derivation is desired to be effectively found as one half of the induction step, obviously $(\mathcal{D}_1)^{(2)}_{*-}$ can be chosen. And for the derivation $\mathcal{D}^{(1)}_{*-}$ of the respective form in (8.13) we can choose the derivation

$$\dfrac{\dfrac{\text{(FOLD/UNFOLD)}}{\mu\tilde{\alpha}.\,\tilde{\tau} = \tilde{\tau}[\mu\tilde{\alpha}.\,\tilde{\tau}/\tilde{\alpha}]} \qquad \begin{array}{c}(\mathcal{D}_1)^{(1)}_{*-}\\ \tilde{\tau}[\mu\tilde{\alpha}.\,\tilde{\tau}/\tilde{\alpha}] \;=\; \chi[\tau_1/\alpha_1,\ldots,\tau_n/\alpha_n]\end{array}}{\mu\tilde{\alpha}.\,\tilde{\tau} = \chi[\tau_1/\alpha_1,\ldots,\tau_n/\alpha_n]}\,\text{TRANS}$$

in $\mathbf{AC}^{=}_{*-}$ without assumption classes. – In the cases of an application of a rule $(\mu{-}\bot)^{\bot\mathrm{der}}_{l}$ or $(\mu{-}\bot)^{\bot\mathrm{der}}_{r}$ at the bottom of $\mathcal{D}$, the presence in $\mathbf{AC}^{=}_{*-}$ of the scheme $(\mu{-}\bot)'$ of axioms of the form $\mu\alpha\alpha_1\ldots\alpha_n.\,\alpha = \bot$, where $\alpha, \alpha_1, \ldots, \alpha_n \in \mu Tp$, is used.

If the last rule application in $\mathcal{D}$ is that of an ARROW-rule, then $\mathcal{D}$ can be written as of the form

$$\dfrac{\begin{array}{c}\big\{[\alpha'_{i_1} : \tau'_{i_1} = \sigma'_{i_1}]^{\alpha'_{i_1}}\big\}_{i_1}\\ \big\{[\alpha'''_{i_3} : \tau'''_{i_3} = \sigma'''_{i_3}]^{\alpha'''_{i_3}}\big\}_{i_3}\\ \mathcal{D}_1\\ \chi_1 : \tilde{\tau}_1 = \tilde{\sigma}_1\end{array} \qquad \begin{array}{c}\big\{[\alpha''_{i_2} : \tau''_{i_2} = \sigma''_{i_2}]^{\alpha''_{i_2}}\big\}_{i_2}\\ \big\{[\alpha'''_{i_3} : \tau'''_{i_3} = \sigma'''_{i_3}]^{\alpha'''_{i_3}}\big\}_{i_3}\\ \mathcal{D}_2\\ \chi_2 : \tilde{\tau}_2 = \tilde{\sigma}_2\end{array}}{\chi_1 \to \chi_2 : \tilde{\tau}_1 \to \tilde{\tau}_2 = \tilde{\sigma}_1 \to \tilde{\sigma}_2}\,\text{ARROW}$$

for some $\tilde{\tau}_1, \tilde{\tau}_2, \tilde{\sigma}_1, \tilde{\sigma}_2, \chi_1, \chi_2 \in \mu Tp$ such that $\tau \equiv \tau_1 \to \tau_2$, $\sigma \equiv \sigma_1 \to \sigma_2$; the indices i_1, i_2 and i_3 in the three kinds of families of marked assumptions indicated at the top range, for some $n_1, n_2, n_3 \in \omega$, over the sets $\{1, \ldots, n_1\}$, $\{1, \ldots, n_2\}$, and $\{1, \ldots, n_3\}$, respectively; hereby $\big\{[\alpha'_{i_1} : \tau'_{i_1} = \sigma'_{i_1}]^{\alpha'_{i_1}}\big\}_{i_1}$ denotes the family of all those open assumption classes in $\mathcal{D}_1$ that are inhabited only in $\mathcal{D}_1$ (but not in $\mathcal{D}_2$), accordingly $\big\{[\alpha''_{i_2} : \tau''_{i_2} = \sigma''_{i_2}]^{\alpha''_{i_2}}\big\}_{i_2}$ denotes the family of all those open assumption classes in $\mathcal{D}_2$ that are inhabited only in $\mathcal{D}_2$ (but not in $\mathcal{D}_1$), and the respective families $\{[\alpha'''_{i_3} : \tau'''_{i_3} = \sigma'''_{i_3}]^{\alpha'''_{i_3}}\}_{i_3}$ indicated at the top of $\mathcal{D}_1$ and of $\mathcal{D}_2$ denote the family of parts in $\mathcal{D}_1$ and $\mathcal{D}_2$ of all those open assumption classes of $\mathcal{D}$ that are inhabited both in $\mathcal{D}_1$ and in $\mathcal{D}_2$. Given this denotation for the open assumption classes in $\mathcal{D}$, Lemma 8.1.4, (a), implies that all type variables α'_{i_1}, α''_{i_2} and α'''_{i_3} occurring as annotations for open marked assumption in $\mathcal{D}$ are mutually different. Since $\mathcal{D}$ was also assumed to be of the form (8.12), we find that

$$\{\alpha_1, \ldots, \alpha_n\} \;=\; \{\alpha'_1, \ldots, \alpha'_{n_1}\} \uplus \{\alpha''_1, \ldots, \alpha''_{n_2}\} \uplus \{\alpha'''_1, \ldots, \alpha'''_{n_3}\} \qquad (8.15)$$

holds and that the family $\big\{[\alpha_i : \tau_i = \sigma_i]^{\alpha_i}\big\}_i$ of open marked assumption classes in $\mathcal{D}$ is the disjoint union of the three families of assumption classes

$\big\{[\alpha'_{i_1} : \tau'_{i_1} = \sigma'_{i_1}]^{\alpha'_{i_1}}\big\}_{i_1}$, $\big\{[\alpha''_{i_2} : \tau''_{i_2} = \sigma''_{i_2}]^{\alpha''_{i_2}}\big\}_{i_2}$ and $\{[\alpha'''_{i_3} : \tau'''_{i_3} = \sigma'''_{i_3}]^{\alpha'''_{i_3}}\}_{i_3}$ appearing above. Furthermore it follows that, for all $i \in \{1, \dots, n\}$, the type variables α_i are distinct.

An application of the induction hypothesis to $\mathcal{D}_1$ implies that from $\mathcal{D}_1$ an derivation $(\mathcal{D}_1)^{(1)}_{*-}$ in $\mathbf{AC}^=_{*-}$ that is of the form

$$\begin{array}{l} (\mathcal{D}_1)^{(1)}_{*-} \\ \tilde{\tau}_1 = \chi_1[\dots, \tau'_{i_1}/\alpha'_{i_1}, \dots, \tau'''_{i_3}/\alpha'''_{i_3}, \dots] \end{array}$$

and that does not contain assumptions can effectively be produced. Let $(\mathcal{D}_1)^{(1)}_{*-}$ be chosen in this way. From the way in which the assumption classes of $\mathcal{D}$ have been denoted above, it follows with the help of Lemma 8.1.4 that

$$\alpha''_{i_2} \notin \mathrm{fv}(\chi_1) \quad \text{for all } i_2 \in \{1, \dots, n_2\} \ .$$

Hence $(\mathcal{D}_1)^{(1)}_{*-}$ can also be written as of the form

$$\begin{array}{l} (\mathcal{D}_1)^{(1)}_{*-} \\ \tilde{\tau}_1 = \underbrace{\chi_1[\dots, \tau'_{i_1}/\alpha'_{i_1}, \dots, \tau''_{i_2}/\alpha''_{i_2}, \dots, \tau'''_{i_3}/\alpha'''_{i_3}, \dots]}_{\underset{(*)}{\equiv} \chi_1[\tau_1/\alpha_1, \dots, \tau_n/\alpha_n]} \end{array}$$

where for the equivalence labeled by (*) it was appealed to (8.15) as well as to the above noted facts, that all α_i for $i \in \{1, \dots, n\}$ are different, and that the family $\big\{[\alpha_i : \tau_i = \sigma_i]^{\alpha_i}\big\}_i$ of open assumption classes in $\mathcal{D}$ is composed by the three families of mutually disjoint assumption classes used above.

By a completely analogous argument it is possible to conclude that an derivation $(\mathcal{D}_2)^{(1)}_{*-}$ in $\mathbf{AC}^=_{*-}$ without assumptions that is of the form

$$\begin{array}{l} (\mathcal{D}_2)^{(1)}_{*-} \\ \tilde{\tau}_2 = \chi_2[\tau_1/\alpha_1, \dots, \tau_n/\alpha_n] \end{array}$$

can effectively be built from $\mathcal{D}_2$; we also choose $(\mathcal{D}_2)^{(2)}_{*-}$ in such a way. From $(\mathcal{D}_1)^{(1)}_{*-}$ and $(\mathcal{D}_2)^{(1)}_{*-}$ we can now build the derivation

$$\dfrac{\begin{array}{c} (\mathcal{D}_1)^{(1)}_{*-} \\ \tilde{\tau}_1 \ = \ \chi_1[\tau_1/\alpha_1, \dots, \tau_n/\alpha_n] \end{array} \qquad \begin{array}{c} (\mathcal{D}_2)^{(1)}_{*-} \\ \tilde{\tau}_2 \ = \ \chi_2[\tau_1/\alpha_1, \dots, \tau_n/\alpha_n] \end{array}}{\tilde{\tau}_1 \to \tilde{\tau}_2 = (\chi_1 \to \chi_2)[\tau_1/\alpha_1, \dots, \tau_n/\alpha_n]} \ \text{ARROW}$$

in $\mathbf{AC}^=_{*-}$; this derivation does not contain assumptions and can be chosen as the derivation $\mathcal{D}^{(1)}_{*-}$ that was desired for the induction step here. The derivation $\mathcal{D}^{(2)}_{*-}$, which is also required for the induction step, can be built in a completely analogous way from the $\mathbf{AC}^{=}_{*-}$-derivations $(\mathcal{D}_1)^{(2)}_{*-}$ and $(\mathcal{D}_2)^{(2)}_{*-}$, whose effective existence is also guaranteed by the induction hypothesis.

If the last rule application in $\mathcal{D}$ is that of an ARROW/FIX-rule, then $\mathcal{D}$ is of the form

$$\dfrac{\begin{array}{c}[\alpha_0 : \tilde{\tau}_1 \to \tilde{\tau}_2 = \tilde{\sigma}_1 \to \tilde{\sigma}_2]^{\alpha_0} \\ \left\{[\alpha_i : \tau_i = \sigma_i]^{\alpha_i}\right\}_i \\ \mathcal{D}_1 \\ \chi_1 : \tilde{\tau}_1 = \tilde{\sigma}_1\end{array} \qquad \begin{array}{c}[\alpha_0 : \tilde{\tau}_1 \to \tilde{\tau}_2 = \tilde{\sigma}_1 \to \tilde{\sigma}_2]^{\alpha_0} \\ \left\{[\alpha_i : \tau_i = \sigma_i]^{\alpha_i}\right\}_i \\ \mathcal{D}_2 \\ \chi_2 : \tilde{\tau}_2 = \tilde{\sigma}_2\end{array}}{\mu\alpha_0.\,(\chi_1 \to \chi_2) : \tilde{\tau}_1 \to \tilde{\tau}_2 = \tilde{\sigma}_1 \to \tilde{\sigma}_2}\ \text{ARROW/FIX}, \alpha_0$$

for some $\tilde{\tau}_1, \tilde{\tau}_2, \tilde{\sigma}_1, \tilde{\sigma}_2, \chi_1, \chi_2 \in \mu Tp$ such that $\tau \equiv \tau_1 \to \tau_2$, $\sigma \equiv \sigma_1 \to \sigma_2$, and for some type variable α_0; the index i varies over the set $\{1, \ldots, n\}$, for some $n \in \omega$; the open assumption class $[\alpha_0 : \tilde{\tau}_1 \to \tilde{\tau}_2 = \tilde{\sigma}_1 \to \tilde{\sigma}_2]^{\alpha_0}$ and the family $\left\{[\alpha_i : \tau_i = \sigma_i]^{\alpha_i}\right\}_i$ of open assumptions classes, where i varies through $i \in \{1, \ldots, n\}$, at the top of $\mathcal{D}_1$ and $\mathcal{D}_2$ stand respectively for *that* parts of these assumption classes in $\mathcal{D}_1$ and accordingly in $\mathcal{D}_2$ which consist of all occurrences of the respective marked assumptions in $\mathcal{D}_1$ and accordingly in $\mathcal{D}_2$; furthermore each of the assumption classes $[\alpha_0 : \tilde{\tau}_1 \to \tilde{\tau}_2 = \tilde{\sigma}_1 \to \tilde{\sigma}_2]^{\alpha_0}$, and $[\alpha_i : \tau_i = \sigma_i]^{\alpha_i}$, for $i \in \{1, \ldots, n\}$, are assumed to be inhabited in $\mathcal{D}$ (but are not assumed to be inhabited necessarily in *both* $\mathcal{D}_1$ and in $\mathcal{D}_2$): for the assumption class $[\alpha_0 : \tilde{\tau}_1 \to \tilde{\tau}_2 = \tilde{\sigma}_1 \to \tilde{\sigma}_2]^{\alpha_0}$ this is due to the side-condition **I** on the bottommost application of ARROW/FIX, for the other assumption classes this follows from the fact that $\mathcal{D}$ was assumed to be of the form (8.12).

Now we describe the construction of the derivation $\mathcal{D}^{(2)}_{*-}$ of the respective form in (8.13) that is required for demonstrating the induction step. By the same reasoning as before in the treatment of the case with an ARROW-rule (i.e. by (1) dividing the occurring assumption classes into those that are inhabited respectively (I) only in $\mathcal{D}_1$, (II) only in $\mathcal{D}_2$ and (III) in both $\mathcal{D}_1$ and $\mathcal{D}_2$, and by (2) using the side-condition **S** for the bottommost application of ARROW/FIX in $\mathcal{D}$ and by (3) using the conditions implied for $\mathcal{D}$ by Lemma 8.1.4), we find that the induction hypothesis, applied for $\mathcal{D}_1$ and for $\mathcal{D}_2$, implies the existence and the possibility to generate effectively from $\mathcal{D}_1$ and $\mathcal{D}_2$ two $\mathbf{AC}^{=}_{*-}$-derivations $(\mathcal{D}_1)^{(2)}_{*-}$ and $(\mathcal{D}_2)^{(2)}_{*-}$ that are of the form

$$\begin{array}{c}(\mathcal{D}_j)^{(2)}_{*-} \\ \tilde{\sigma}_j = \chi_j[\tilde{\sigma}_1 \to \tilde{\sigma}_2/\alpha_0, \sigma_1/\alpha_1, \ldots, \sigma_n/\alpha_n]\end{array} \qquad \text{(for each } j \in \{1, 2\}\text{)}.$$

and that do not contain assumptions. Using $(\mathcal{D}_1)^{(2)}_{*-}$ and $(\mathcal{D}_2)^{(2)}_{*-}$, it is now possible to assemble a derivation in $\mathbf{AC}^{=}_{*-}$ of the form

$$\dfrac{\dfrac{\begin{matrix}(\mathcal{D}_1)^{(2)}_{*-} & \qquad & (\mathcal{D}_2)^{(2)}_{*-}\\ \tilde{\sigma}_1 = \chi_1[\tilde{\sigma}_1 \to \tilde{\sigma}_2/\alpha_0, \ldots] & \qquad & \tilde{\sigma}_2 = \chi_2[\tilde{\sigma}_1 \to \tilde{\sigma}_2/\alpha_0, \ldots]\end{matrix}}{\tilde{\sigma}_1 \to \tilde{\sigma}_2 = \underbrace{(\chi_1 \to \chi_2)[\tilde{\sigma}_1 \to \tilde{\sigma}_2/\alpha_0, \sigma_1/\alpha_1, \ldots, \sigma_n/\alpha_n]}_{\underset{(8.16)}{\equiv}\ ((\chi_1\to\chi_2)[\sigma_1/\alpha_1,\ldots,\sigma_n/\alpha_n])[\tilde{\sigma}_1\to\tilde{\sigma}_2/\alpha_0]}}\,\text{ARROW}}{\tilde{\sigma}_1 \to \tilde{\sigma}_2 = \underbrace{\mu\alpha_0.\,((\chi_1 \to \chi_2)[\sigma_1/\alpha_1, \ldots, \sigma_n/\alpha_n])}_{\underset{(8.16)}{\equiv}\ (\mu\alpha_0.\,(\chi_1\to\chi_2))[\sigma_1/\alpha_1,\ldots,\sigma_n/\alpha_n]}}\,\text{CONTRACT}$$

that does not contain assumptions and that can be taken as the desired derivation $\mathcal{D}^{(2)}_{*-}$ (the side-condition $\alpha_0 \downarrow (\chi_1 \to \chi_2)[\sigma_1/\alpha_1, \ldots, \sigma_n/\alpha_n]$ for the displayed application of CONTRACT at the bottom of this derivation is obviously satisfied). At two places we have used here that

$$\alpha_0 \notin \mathrm{fv}(\sigma_j) \quad \text{for } j \in \{1, \ldots, n\} \tag{8.16}$$

holds, which is implied by applications of Lemma 8.1.4, (c), to both $\mathcal{D}_1$ and $\mathcal{D}_2$ and by the side-condition **S** on the bottommost application of ARROW/FIX in $\mathcal{D}$. By having explained how to construct the derivation $\mathcal{D}^{(2)}_{*-}$ in an effective way from the assumed derivation $\mathcal{D}$ we have shown one half of what is necessary to prove for the induction step in this case.

The derivation $\mathcal{D}^{(1)}_{*-}$, whose effective existence is also required to be proved, can be built in an analogous way from the derivations $(\mathcal{D}_1)^{(1)}_{*-}$ and $(\mathcal{D}_2)^{(1)}_{*-}$ that can effectively be constructed in their turn from $\mathcal{D}_1$ and $\mathcal{D}_2$ due to the induction hypothesis.

This concludes the induction on $|\mathcal{D}|$ with $\mathcal{D}$ as in (8.12) for the proof of item (i) of the lemma.

□

The existence of an effective transformation from an arbitrary derivation $\mathcal{D}$ in $\mathbf{HB}_\mathbf{0}^=$ without open assumption classes via an annotated derivation $\hat{\mathcal{D}}$ in $\mathbf{ann\text{-}HB}_\mathbf{0}^=$ into a derivation $(\hat{D})'$ in $\mathbf{AC}^=$ with the same conclusion as $\mathcal{D}$ is now a direct consequence of the statements shown so far (compare also the illustration of the three main steps of this transformation in Figure 8.1).

Theorem 8.1.8. *Every derivation in* $\mathbf{HB}_\mathbf{0}^=$ *without open assumption classes can effectively be transformed into a derivation in* $\mathbf{AC}^=$ *with the same conclusion and without assumptions.*

Proof. This is an immediate consequence of Lemma 8.1.4 and Lemma 8.1.6. □

Now we want to illustrate this transformation by performing a 'test-run' on the example of a $\mathbf{HB}_\mathbf{0}^=$-derivation without open assumptions that we have encountered in Chapter 6 before.

Example 8.1.9. We consider the two strongly equivalent recursive types

$$\tau \equiv \mu\alpha.\,((\alpha \to \alpha) \to \alpha) \quad \text{and} \quad \sigma \equiv \mu\alpha.\,(\alpha \to (\alpha \to \alpha))\ , \tag{8.17}$$

which correspond to the recursive types τ_2 and σ_2 in Example 3.6.3, and a derivation of $\tau = \sigma$ in $\mathbf{HB}_{\mathbf{0}}^=$ that we have encountered before, with different assumption markers, in Figure 6.2. We let $\mathcal{D}$ be the derivation

$$\begin{prooftree}
\AxiomC{$(\ldots)^\beta$}
\doubleLine
\UnaryInfC{$\tau = \sigma$}
\AxiomC{$(\ldots)^\gamma$}
\UnaryInfC{$\tau \to \tau = \sigma$}
\AxiomC{$(\ldots)^\beta$}
\doubleLine
\UnaryInfC{$\tau = \sigma$}
\BinaryInfC{$(\tau \to \tau) \to \tau = \sigma \to \sigma$}
\UnaryInfC{$\tau = \sigma \to \sigma$}
\RightLabel{γ}
\BinaryInfC{$\tau \to \tau = \sigma \to (\sigma \to \sigma)$}
\RightLabel{FOLD$_r$}
\UnaryInfC{$\tau \to \tau = \sigma$}
\AxiomC{$(\ldots)^\beta$}
\doubleLine
\UnaryInfC{$\tau = \sigma$}
\AxiomC{$(\ldots)^\delta$}
\UnaryInfC{$\tau = \sigma \to \sigma$}
\BinaryInfC{$\tau \to \tau = \sigma \to (\sigma \to \sigma)$}
\UnaryInfC{$\tau \to \tau = \sigma$}
\AxiomC{$(\ldots)^\beta$}
\doubleLine
\UnaryInfC{$\tau = \sigma$}
\RightLabel{δ}
\BinaryInfC{$(\tau \to \tau) \to \tau = \sigma \to \sigma$}
\RightLabel{FOLD$_l$}
\UnaryInfC{$\tau = \sigma \to \sigma$}
\RightLabel{(ARROW/FIX)$_\beta$}
\BinaryInfC{$(\tau \to \tau) \to \tau = \sigma \to (\sigma \to \sigma)$}
\doubleLine
\RightLabel{FOLD$_{l/r}$}
\UnaryInfC{$\underbrace{\mu\alpha.\,((\alpha \to \alpha) \to \alpha)}_{\equiv\tau} = \underbrace{\mu\alpha.\,(\alpha \to (\alpha \to \alpha))}_{\equiv\sigma}$}
\end{prooftree}$$

in $\mathbf{HB}_{\mathbf{0}}^=$. The rule applications labeled by γ, and by δ in this symbolic prooftree for $\mathcal{D}$ are applications of ARROW/FIX at which the assumption classes marked by γ, and respectively those marked by δ are discharged. Obviously, the derivation $\mathcal{D}$ contains no open assumption classes. We set out to perform the three steps in the transformation developed in this section and to observe its main effects on the 'input' $\mathcal{D}$. We want to build, in a stepwise manner, a derivation $(\hat{\mathcal{D}})'$ in $\mathbf{AC}^=$ with the same conclusion as $\mathcal{D}$ and without assumptions.

In performing the steps of the transformation developed in this section, we will at some places only display the occurring intermediary derivations in $\mathbf{AC}_{*-}^=$, and will then only hint how these can be transformed further into corresponding $\mathbf{AC}^=$-derivations. We do so because the respective $\mathbf{AC}^=$-derivations tend to become quite large and very hard to deal with typographically, and furthermore because the elimination of applications of FOLD$_{l/r}$ and CONTRACT from derivations in $\mathbf{AC}_{*-}^=$+FOLD$_{l/r}$ with the result of $\mathbf{AC}^=$-derivations with respectively the same conclusion is always a very easy matter (the elimination of CONTRACT-applications has been described in the proof of Proposition 5.1.6).

The first step in the transformation of $\mathcal{D}$, the *annotation step*, consists, along the proof of Lemma 8.1.5, (i), in finding appropriate annotations for the formulas in $\mathcal{D}$ such that the result $\hat{\mathcal{D}}$ of prefixing these annotations in the respective formulas of $\mathcal{D}$ is a derivation in **ann-**$\mathbf{HB}_{\mathbf{0}}^=$. We observe that the markers used for assumptions in $\mathcal{D}$ are already from the start recursive types, which furthermore satisfy the conditions (1) and (2) in item (a) of the proof of Lemma 8.1.5, namely, (1) they do not occur as variables anywhere in formulas of $\mathcal{D}$, and (2) they fulfill the property that different assumption classes in $\mathcal{D}$ are marked by different type variables. Hence we are able to follow the inductive definition of $\hat{\mathcal{D}}$ from $\mathcal{D}$ sketched in item (a) of the proof of

Lemma 8.1.5. In doing so, we arrive at a derivation $\hat{\mathcal{D}}$ of the form

$$
\begin{prooftree}
\AxiomC{$(\beta : \dots)^{\beta}$}
\doubleLine
\UnaryInfC{$\beta : \tau = \sigma$}
\AxiomC{$(\gamma : \dots)^{\gamma}$}
\UnaryInfC{$\gamma : \tau \to \tau = \sigma$}
\AxiomC{$(\beta : \dots)^{\beta}$}
\doubleLine
\UnaryInfC{$\beta : \tau = \sigma$}
\BinaryInfC{$\gamma \to \beta : \;\dots$}
\UnaryInfC{$\gamma \to \beta : \; \tau = \sigma \to \sigma$}
\RightLabel{γ}
\BinaryInfC{$\mu\gamma.\,(\beta \to (\gamma \to \beta)) : \;\dots$}
\UnaryInfC{$\mu\gamma.\,(\beta \to (\gamma \to \beta)) : \; \tau \to \tau = \sigma$}
\AxiomC{$(\beta : \dots)^{\beta}$}
\doubleLine
\UnaryInfC{$\beta : \tau = \sigma$}
\AxiomC{$(\delta : \dots)^{\delta}$}
\UnaryInfC{$\delta : \tau = \sigma \to \sigma$}
\BinaryInfC{$\beta \to \delta : \;\dots$}
\UnaryInfC{$\beta \to \delta : \; \tau \to \tau = \sigma$}
\AxiomC{$(\beta : \dots)^{\beta}$}
\doubleLine
\UnaryInfC{$\beta : \tau = \sigma$}
\RightLabel{δ}
\BinaryInfC{$\mu\delta.\,((\beta \to \delta) \to \beta) : \;\dots$}
\UnaryInfC{$\mu\delta.\,((\beta \to \delta) \to \beta) : \; \tau = \sigma \to \sigma$}
\RightLabel{β}
\BinaryInfC{$\mu\beta.\,\big(\mu\gamma.\,(\beta \to (\gamma \to \beta)) \to \mu\delta.\,((\beta \to \delta) \to \beta)\big) : \; (\tau \to \tau) \to \tau = \sigma \to (\sigma \to \sigma)$}
\doubleLine
\UnaryInfC{$\mu\beta.\,\big(\mu\gamma.\,(\beta \to (\gamma \to \beta)) \to \mu\delta.\,((\beta \to \delta) \to \beta)\big) : \; \tau = \sigma$}
\end{prooftree}
$$

in $\textbf{ann-HB}_{\mathbf{0}}^{=}$, where we have abbreviated, for typographical reasons, the marked assumptions

$$(\beta : \, (\tau \to \tau) \to \tau = \sigma \to (\sigma \to \sigma))^{\beta} \;, \quad (\gamma : \, \tau \to \tau = \sigma \to (\sigma \to \sigma))^{\gamma} \;,$$
$$\text{and} \quad (\delta : \, (\tau \to \tau) \to \tau = \sigma \to \sigma)^{\delta}$$

at the top of $\hat{\mathcal{D}}$ as well as the annotated formulas

$$\gamma \to \beta : \, (\tau \to \tau) \to \tau = \sigma \to \sigma \;, \quad \beta \to \delta : \, \tau \to \tau = \sigma \to (\sigma \to \sigma) \;,$$
$$\mu\gamma.\,(\beta \to (\gamma \to \beta)) : \, \tau \to \tau = \sigma \to (\sigma \to \sigma),$$
$$\text{and} \quad \mu\delta.\,((\beta \to \delta) \to \beta) : \, (\tau \to \tau) \to \tau = \sigma \to \sigma$$

within $\hat{\mathcal{D}}$. We note that, equally as $\mathcal{D}$, the annotated derivation $\hat{\mathcal{D}}$ does not contain open assumption classes (as this also stated for the transformation developed in the proof of Lemma 8.1.5, (i), by the assertion of the Precise Formulation of this statement; see also Figure 8.3).

The second step of the transformation, the *extraction step*, consists in the application of the transformation guaranteed by Lemma 8.1.6, (i), which implies that two derivations $\hat{\mathcal{D}}^{(1)}$ and $\hat{\mathcal{D}}^{(2)}$ in $\mathbf{AC}^{=}$ with the respective conclusions $\tau = \chi$ and $\sigma = \chi$ can effectively be extracted from the $\textbf{ann-HB}_{\mathbf{0}}^{=}$-derivation $\hat{\mathcal{D}}$, where we designate with χ, here and below, the annotation in the conclusion of $\hat{\mathcal{D}}$, i.e. we let

$$\chi \equiv \; \mu\beta.\,\big(\mu\gamma.\,(\beta \to (\gamma \to \beta)) \to \mu\delta.\,((\beta \to \delta) \to \beta)\big) \;. \tag{8.18}$$

In following the proof of Lemma 8.1.6, we find that we first have to extract from $\hat{\mathcal{D}}$ two derivations $(\hat{\mathcal{D}})^{(1)}_{*-}$ and $(\hat{\mathcal{D}})^{(2)}_{*-}$ in $\mathbf{AC}^{=}_{*-}$ without assumptions and with the respective conclusions $\tau = \chi$ and $\chi = \rho$, which derivations have to be translated subsequently into the desired $\mathbf{AC}^{=}$-derivations $\hat{\mathcal{D}}^{(1)}$ and $\hat{\mathcal{D}}^{(2)}$.

For the purpose of being able to deal with the arising derivations typographically, we use the following abbreviating notation: we denote by $\hat{\mathcal{D}}_{111}$, and respectively by $\hat{\mathcal{D}}_{112}$ those two subderivations of $\hat{\mathcal{D}}$ that lead up to the left, and to the right premise of *that* application of ARROW/FIX in $\hat{\mathcal{D}}$ at which the assumptions marked by β are discharged. Thus $\hat{\mathcal{D}}_{111}$ is of the form

$$\dfrac{\dfrac{(\,\beta\colon(\tau\to\tau)\to\tau=\sigma\to(\sigma\to\sigma)\,)^{\beta}}{\beta:\tau=\sigma} \qquad \dfrac{\dfrac{\dfrac{(\,\gamma\colon\tau\to\tau=\sigma\to(\sigma\to\sigma)\,)^{\gamma}}{\gamma:\tau\to\tau=\sigma} \quad \dfrac{(\,\beta\colon(\tau\to\tau)\to\tau=\sigma\to(\sigma\to\sigma)\,)^{\beta}}{\beta:\tau=\sigma}}{\gamma\to\beta:(\tau\to\tau)\to\tau=\sigma\to\sigma}\,\text{ARROW}}{\gamma\to\beta:\tau=\sigma\to\sigma}\,\text{FOLD}_{l}}{\dfrac{\mu\gamma.\,(\beta\to(\gamma\to\beta)):\tau\to\tau=\sigma\to(\sigma\to\sigma)}{\mu\gamma.\,(\beta\to(\gamma\to\beta)):\tau\to\tau=\sigma}\,\text{FOLD}_{r}}\,\text{ARROW/FIX},\,\gamma \tag{8.19}$$

and has $[\beta:(\tau\to\tau)\to\tau=\sigma\to(\sigma\to\sigma)]^{\beta}$ as its single class of open assumptions (the assumption with marker γ has already been discharged in $\hat{\mathcal{D}}_{111}$). The subderivation $\hat{\mathcal{D}}_{112}$ of $\hat{\mathcal{D}}$ is given below in (8.21). These two subderivations of $\hat{\mathcal{D}}$ are now displayed in (8.19), and respectively in (8.21), in greater detail than this was typographically possible in the earlier given, abbreviated prooftree for $\hat{\mathcal{D}}$.

For extracting the two needed $\mathbf{AC}^{=}_{*-}$-derivations $(\hat{\mathcal{D}})^{(1)}_{*-}$ and $(\hat{\mathcal{D}})^{(2)}_{*-}$ from $\hat{\mathcal{D}}$, we can effect the transformation that is defined in the proof of Lemma 8.1.7, (i), by induction on the depth of an arbitrary $\textbf{ann-HB}^{=}_{\mathbf{0}}$-derivations. Here it follows that in the course of building the derivation $(\hat{\mathcal{D}})^{(1)}_{*-}$ from $\hat{\mathcal{D}}$, we encounter the subtask of having to produce the derivation $(\hat{\mathcal{D}}_{111})^{(1)}_{*-}$ from $\hat{\mathcal{D}}_{111}$ at some earlier stage, and that $(\hat{\mathcal{D}}_{111})^{(1)}_{*-}$ is bound to appear eventually in the result $(\hat{\mathcal{D}})^{(1)}_{*-}$; and a similar assertion holds for $(\hat{\mathcal{D}}_{112})^{(1)}_{*-}$ with respect to $\hat{\mathcal{D}}_{112}$ and $\hat{\mathcal{D}}^{(1)}_{*-}$. By following the inductive definition described in the proof of Lemma 8.1.7, (i), we arrive, as can be verified in a straightforward manner, at the derivation $(\hat{\mathcal{D}}_{111})^{(1)}_{*-}$ in $\mathbf{AC}^{=}_{*-}$ that results from the derivation

$$\dfrac{\dfrac{\dfrac{\dfrac{\text{(REFL)}}{(\tau\to\tau)\to\tau=\beta\,[(\tau\to\tau)\to\tau/\beta]}}{\tau=\beta\,[(\tau\to\tau)\to\tau/\beta]}\,\text{FOLD}_{l} \qquad \dfrac{\dfrac{\dfrac{\text{(REFL)}}{\tau\to\tau=\gamma\,[\tau\to\tau/\gamma]} \quad \dfrac{\dfrac{\text{(REFL)}}{(\tau\to\tau)\to\tau=\beta\,[(\tau\to\tau)\to\tau/\beta]}}{\tau=\beta\,[(\tau\to\tau)\to\tau/\beta]}\,\text{FOLD}_{l}}{(\tau\to\tau)\to\tau=(\gamma\to\beta)\,[(\tau\to\tau)\to\tau/\beta,\tau\to\tau/\gamma]}\,\text{ARROW}}{\tau=(\gamma\to\beta)\,[(\tau\to\tau)\to\tau/\beta,\tau\to\tau/\gamma]}\,\text{FOLD}_{l}}{\tau\to\tau=\underbrace{(\beta\to(\gamma\to\beta))\,[(\tau\to\tau)\to\tau/\beta,\tau\to\tau/\gamma]}_{\equiv\,((\beta\to(\gamma\to\beta))\,[(\tau\to\tau)\to\tau/\beta])\,[\tau\to\tau/\gamma]}}\,\text{ARROW}$$

$$\dfrac{\qquad\qquad}{\tau\to\tau=\underbrace{\mu\gamma.\,\big((\beta\to(\gamma\to\beta))\,[(\tau\to\tau)\to\tau/\beta]\big)}_{\equiv\,(\mu\gamma.\,(\beta\to(\gamma\to\beta))\,[(\tau\to\tau)\to\tau/\beta])}}\,\text{CONTRACT} \tag{8.20}$$

in $\mathbf{AC}^{=}_{*-}$+FOLD$_l$ by eliminating each of the three applications of FOLD$_l$, in each case using instead an axiom (FOLD/UNFOLD) that is followed by an application of TRANS.

In a similar way as $(\hat{\mathcal{D}}_{111})^{(1)}_{*-}$ has been extracted inductively from the subderivation $\hat{\mathcal{D}}_{111}$ of $\hat{\mathcal{D}}$, from the subderivation $\hat{\mathcal{D}}_{112}$ of $\mathcal{D}$ that is of the form

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{(\beta\colon(\tau\to\tau)\to\tau=\sigma\to(\sigma\to\sigma))^{\beta}}{\overline{\beta : \tau=\sigma}} \qquad \dfrac{(\delta\colon(\tau\to\tau)\to\tau=\sigma\to\sigma)^{\delta}}{\delta : \tau=\sigma\to\sigma}\,\mathrm{FOLD}_l}{\beta\to\delta : \tau\to\tau=\sigma\to(\sigma\to\sigma)}\,\mathrm{ARROW}}{\beta\to\delta : \tau\to\tau=\sigma}\,\mathrm{FOLD}_r \qquad \dfrac{(\beta\colon(\tau\to\tau)\to\tau=\sigma\to(\sigma\to\sigma))^{\beta}}{\overline{\beta : \tau=\sigma}}}{\mu\delta.\,((\beta\to\delta)\to\beta) : (\tau\to\tau)\to\tau=\sigma\to\sigma}\,\mathrm{ARROW/FIX},\,\delta}{\mu\delta.\,((\beta\to\delta)\to\beta) : \tau=\sigma\to\sigma}\,\mathrm{FOLD}_l \tag{8.21}$$

a derivation $(\hat{\mathcal{D}}_{112})^{(1)}_{*-}$ in $\mathbf{AC}^=_{*-}$ can be extracted that results from the derivation

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\text{(REFL)}}{(\tau\to\tau)\to\tau=\beta\,[(\tau\to\tau)\to\tau/\beta]}}{\tau=\beta\,[(\tau\to\tau)\to\tau/\beta]} \qquad \dfrac{\dfrac{\text{(REFL)}}{(\tau\to\tau)\to\tau=\delta\,[(\tau\to\tau)\to\tau/\delta]}}{\tau=\delta\,[(\tau\to\tau)\to\tau/\delta]}}{\tau\to\tau=(\beta\to\delta)[(\tau\to\tau)\to\tau/\beta,(\tau\to\tau)\to\tau/\delta]}\,\mathrm{ARROW} \qquad \dfrac{\dfrac{\text{(REFL)}}{(\tau\to\tau)\to\tau=\beta\,[(\tau\to\tau)\to\tau/\beta]}}{\tau=\beta\,[(\tau\to\tau)\to\tau/\beta]}}{(\tau\to\tau)\to\tau=\underbrace{((\beta\to\delta)\to\beta)\,[(\tau\to\tau)\to\tau/\beta,(\tau\to\tau)\to\tau/\delta]}_{\equiv\,(\,((\beta\to\delta)\to\beta)\,[(\tau\to\tau)\to\tau/\beta]\,)\,[(\tau\to\tau)\to\tau/\delta]}}\,\mathrm{ARROW}}{(\tau\to\tau)\to\tau=\underbrace{\mu\delta.\,\big(((\beta\to\delta)\to\beta)\,[(\tau\to\tau)\to\tau/\beta]\big)}_{\equiv\,(\,\mu\delta.\,((\beta\to\delta)\to\beta)\,)\,[(\tau\to\tau)\to\tau/\beta]}}\,\mathrm{CONTRACT}}{\tau=\big(\mu\delta.\,((\beta\to\delta)\to\beta)\big)\,[(\tau\to\tau)\to\tau/\beta]}\,\mathrm{FOLD}_l \tag{8.22}$$

in $\mathbf{AC}^=_{*-}$+FOLD$_l$ by eliminating each of the four occurring applications of FOLD$_l$ analogously as indicated above for the case of the derivation in (8.20).

By continuing to follow the inductive construction of $(\hat{\mathcal{D}})^{(1)}_{*-}$ from $\hat{\mathcal{D}}$ it is possible to verify that, relative to the already reached derivations $\hat{\mathcal{D}}_{111}$ and $\hat{\mathcal{D}}_{112}$, the derivation $(\hat{\mathcal{D}})^{(1)}_{*-}$ in $\mathbf{AC}^=_{*-}$ results from the derivation

$$\dfrac{\dfrac{\dfrac{\begin{array}{c}(\hat{\mathcal{D}}_{111})^{(1)}_{*-}\\ \tau\to\tau=(\mu\gamma.\,(\beta\to(\gamma\to\beta)))\,[(\tau\to\tau)\to\tau/\beta]\end{array} \qquad \begin{array}{c}(\hat{\mathcal{D}}_{112})^{(1)}_{*-}\\ \tau=(\mu\delta.\,((\beta\to\delta)\to\beta))\,[(\tau\to\tau)\to\tau/\beta]\end{array}}{(\tau\to\tau)\to\tau=(\mu\gamma.\,(\beta\to(\gamma\to\beta))\to\mu\delta.\,((\beta\to\delta)\to\beta))[(\tau\to\tau)\to\tau/\beta]}\,\mathrm{ARROW}}{(\tau\to\tau)\to\tau=\mu\beta.\,\big(\mu\gamma.\,(\beta\to(\gamma\to\beta))\to\mu\delta.\,((\beta\to\delta)\to\beta)\big)}\,\mathrm{CONTRACT}}{\tau=\mu\beta.\,\big(\mu\gamma.\,(\beta\to(\gamma\to\beta))\to\mu\delta.\,((\beta\to\delta)\to\beta)\big)}\,\mathrm{FOLD}_l \tag{8.23}$$

by eliminating (as hinted above) the bottommost application of FOLD$_l$.

Having obtained the $\mathbf{AC}^=_{*-}$-derivation $(\hat{\mathcal{D}})^{(1)}_{*-}$, it can be transformed into the derivation $(\hat{\mathcal{D}})^{(1)}$ in $\mathbf{AC}^=$ required for the extraction step, by the transformation stated by Corollary 7.1.17, and hence by eliminating the three applications of CONTRACT in $(\hat{\mathcal{D}})^{(1)}_{*-}$ (that can be seen in (8.20), (8.22) and (8.23), respectively) using the statement of Proposition 5.1.6.

The construction of the second derivation needed for the extraction step, the derivation $(\hat{\mathcal{D}})^{(2)}$ in $\mathbf{AC}^=$ with conclusion $\sigma=\chi$ and without assumptions can be performed in an analogous manner: first a derivation $(\hat{\mathcal{D}})^{(2)}_{*-}$ in $\mathbf{AC}^{=}_{*-}$ without assumptions and with conclusion $\sigma = \chi$ is extracted from $\hat{\mathcal{D}}$ according to the proof of Lemma 8.1.7, (i), and then $(\hat{\mathcal{D}})^{(2)}_{*-}$ is transformed into the desired derivation $(\hat{\mathcal{D}})^{(2)}$ in $\mathbf{AC}^{=}$ by eliminating CONTRACT-applications from $(\hat{\mathcal{D}})^{(2)}_{*-}$ according to the transformation stated by Proposition 5.1.6. Since the extraction of $(\hat{\mathcal{D}})^{(2)}$ in this way from $\hat{\mathcal{D}}$ is straightforward and similar to the extraction of $(\hat{\mathcal{D}})^{(1)}$ from $\hat{\mathcal{D}}$ outlined above, we do not show it here. We only stipulate $\hat{\mathcal{D}}^{(2)}_{*-}$ to be the $\mathbf{AC}^{=}$-derivation with conclusion $\sigma = \chi$ and without assumptions that is found in the way sketched here.

In the third step of the transformation, the *combination step*, the two reached derivations $(\hat{\mathcal{D}})^{(1)}$ and $(\hat{\mathcal{D}})^{(2)}$ in $\mathbf{AC}^{=}$ with the respective conclusions $\tau = \chi$ and $\sigma = \chi$ (where χ was defined in (8.18)) are combined with the $\mathbf{AC}^{=}$-derivation $(\hat{\mathcal{D}})'$

$$\dfrac{\begin{array}{c}(\hat{\mathcal{D}})^{(1)}\\ \tau = \mu\beta.\,(\mu\gamma.\,(\beta\to(\gamma\to\beta))\to\mu\delta.\,((\beta\to\delta)\to\beta))\end{array} \qquad \dfrac{\begin{array}{c}(\hat{\mathcal{D}})^{(2)}\\ \sigma = \mu\beta.\,(\mu\gamma.\,(\beta\to(\gamma\to\beta))\to\mu\delta.\,((\beta\to\delta)\to\beta))\end{array}}{\mu\beta.\,(\mu\gamma.\,(\beta\to(\gamma\to\beta))\to\mu\delta.\,((\beta\to\delta)\to\beta)) = \sigma}\,\text{SYMM}}{\underbrace{\mu\alpha.\,((\alpha\to\alpha)\to\alpha)}_{\equiv\tau} = \underbrace{\mu\alpha.\,(\alpha\to(\alpha\to\alpha))}_{\equiv\sigma}}\,\text{TRANS}$$

as the result that does not contain assumptions. This step followed the assertion of Lemma 8.1.6, (ii), and is a transformation similar to the one described in the proof of Lemma 8.1.7, (ii). It is easy to see that the derivation $(\hat{\mathcal{D}})'$ is actually also a derivation in the μ-COMPAT-free variant system $\mathbf{AC}^{=}_{-}$ of $\mathbf{AC}^{=}$.

In this way we have outlined how to construct, from the given derivation $\mathcal{D}$ in $\mathbf{HB}^{=}_{\mathbf{0}}$ without open assumptions, a derivation $(\hat{\mathcal{D}})'$ in $\mathbf{AC}^{=}$ without assumptions and with the same conclusion as $\mathcal{D}$.

■

In conclusion of this section we want to report of some of our attempts to generalize the transformation developed here from $\mathbf{HB}^{=}_{\mathbf{0}}$-derivations into $\mathbf{AC}^{=}$-derivations.

Remark 8.1.10 (Generalizations of the transformation from $\mathbf{HB}^{=}_{\mathbf{0}}$ to $\mathbf{AC}^{=}$). In the below items (a) and (b) we consider respectively the questions of whether the transformation from derivations in $\mathbf{HB}^{=}_{\mathbf{0}}$ into derivations in $\mathbf{AC}^{=}$ constructed in this section can be generalized to one from $\mathbf{e}$-$\mathbf{HB}^{=}_{\mathbf{0}}$- into $\mathbf{AC}^{=}$-derivations, or to one from $\mathbf{HB}^{=}$- into $\mathbf{AC}^{=}$-derivations.

(a) The transformation developed in this section from $\mathbf{HB}^{=}_{\mathbf{0}}$-derivations via derivations in $\mathbf{ann}$-$\mathbf{HB}^{=}_{\mathbf{0}}$, and in fact via $\mathbf{AC}^{=}_{*-}$-derivations, into $\mathbf{AC}^{=}$-derivations can actually be extended, in a rather straightforward way, into one that transforms $\mathbf{e}$-$\mathbf{HB}^{=}_{\mathbf{0}}$-derivations into $\mathbf{AC}^{=}$-derivations in an analogous manner. A prerequisite for such a transformation is an annotated version $\mathbf{ann}$-$\mathbf{e}$-$\mathbf{HB}^{=}_{\mathbf{0}}$ of the extension $\mathbf{e}$-$\mathbf{HB}^{=}_{\mathbf{0}}$ of $\mathbf{HB}^{=}_{\mathbf{0}}$ which is easy to define (cf. Remark 8.1.2, (b), for a definition of a generalized FIX-rule, on which the introduction of such an annotated system **ann-e-**$\mathbf{HB}_0^=$ can be based). Relying on such a system **ann-e-**$\mathbf{HB}_0^=$, it is then straightforward to define a transformation from **e-**$\mathbf{HB}_0^=$-derivations into $\mathbf{AC}^=$-derivations that proceeds by the same kind of steps as are depicted in Figure 8.1: a given derivation $\mathcal{D}$ in **e-**$\mathbf{HB}_0^=$ without open assumptions and with conclusion $\tau = \sigma$, for some $\tau, \sigma \in \mu Tp$, is first *annotated* into a derivation $\hat{\mathcal{D}}$ in **ann-e-**$\mathbf{HB}_0^=$ without open assumptions and with conclusion $\chi : \tau = \sigma$, for some $\chi \in \mu Tp$; then two $\mathbf{AC}^=$-derivations $(\hat{\mathcal{D}})^{(1)}$ and $(\hat{\mathcal{D}})^{(2)}$ (or for that matter, $\mathbf{AC}^=_{*-}$-derivations $(\hat{\mathcal{D}})^{(1)}_{*-}$ and $(\hat{\mathcal{D}})^{(2)}_{*-}$) without assumptions and with respective conclusion $\tau = \chi$ and $\sigma = \chi$ are *extracted* from $\hat{\mathcal{D}}$; and finally, $(\hat{\mathcal{D}})^{(1)}$ and $(\hat{\mathcal{D}})^{(2)}$ are *combined* into an $\mathbf{AC}^=$-derivation $(\hat{\mathcal{D}})'$ in $\mathbf{AC}^=$ with the same conclusion as $\mathcal{D}$ and without assumptions.

(b) It is not clear to us at present how the transformation developed in this section could be generalized into one that is applicable to arbitrary derivations $\mathcal{D}$ in $\mathbf{HB}^=$ without open assumptions (and not just to $\mathbf{HB}^=$-derivations of special form in the sense of fn. 1 on page 1), that proceeds in the spirit of the transformation given here, and that produces a derivation $\mathcal{D}'$ in $\mathbf{AC}^=$ without assumptions and with the same conclusion as $\mathcal{D}$. We will also pose this as Open Problem 9.1.1 in Chapter 9.

8.2 A Transformation of Derivations in $\mathbf{HB}^=$ into Derivations in $\mathbf{HB}_0^=$

The existence of a transformation from derivations in $\mathbf{HB}_0^=$ into derivations in $\mathbf{HB}^=$ that preserves respective conclusions and open assumption classes was stated in Lemma 5.1.19, Chapter 5; in the proof of this lemma we described an easy transformation to this effect. We have also mentioned that it is not equally simple to give also a transformation for the opposite direction, i.e. one that is able to produce, for every given derivation in $\mathbf{HB}^=$, a derivation in $\mathbf{HB}_0^=$ with the same conclusion and with the same open assumption classes.

In fact, whereas the transformation given in the proof of Lemma 5.1.19 acts on all derivations $\mathcal{D}$ in $\mathbf{HB}_0^=$ *with or without open assumption classes* and produces a derivation $\mathcal{D}'$ in $\mathbf{HB}^=$ with the same conclusion and with the same open assumption classes, a transformation in the opposite direction with an analogous property does not exist in general. So it is, for instance, the case that for none of the two derivations consisting of just the applications

$$\frac{(\bot = \alpha)^u}{\alpha = \bot}\,\text{SYMM} \qquad \text{and} \qquad \frac{(\bot = \alpha)^v \qquad (\alpha = \top)^w}{\bot = \top}\,\text{TRANS}$$

of SYMM and TRANS (both of which derivations contain open marked assumptions) there exists a respective mimicking derivation in $\mathbf{HB}_0^=$, i.e. a derivation with respectively the same conclusion and with the same open assumption classes. In both cases this is a simple consequence of the fact that the system $\mathbf{HB}_0^=$ fulfills the subformula property SP_1 (cf. Proposition 5.1.17, Chapter 5, Section 5.1): due to $\alpha \not\sqsubseteq \bot$ (and also due to $\bot \not\sqsubseteq \alpha$), there does not exist a derivation in $\mathbf{HB}_{\mathbf{0}}^{=}$ with conclusion $\alpha = \bot$ and with the formula $\bot = \alpha$ occurring in an open assumption; similarly, there does not exist *such* a $\mathbf{HB}_{\mathbf{0}}^{=}$-derivation with conclusion $\bot = \top$ *that* contains an open assumption $\bot = \alpha$ (or $\alpha = \bot$). This observation can also be reformulated as the following proposition.

Proposition 8.2.1. *The rules* SYMM *and* TRANS *are not derivable in* $\mathbf{HB}_{\mathbf{0}}^{=}$.

Therefore a general transformation which would be able to transform *every* derivation $\mathcal{D}$ in $\mathbf{HB}_{\mathbf{0}}^{=} + \mathrm{SYMM} + \mathrm{TRANS}$ into a derivation $\mathcal{D}'$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ that mimics $\mathcal{D}$ *does not exist*.

However, if we restrict our considerations to derivations in $\mathbf{HB}^{=}$ *without open assumptions*, then there does certainly exist an effective transformation of derivations $\mathcal{D}$ into respective mimicking derivations in $\mathbf{HB}_{\mathbf{0}}^{=}$, i.e. into derivations in $\mathbf{HB}_{\mathbf{0}}^{=}$ without open assumptions and with respectively the same conclusion. As we have already mentioned at the outset of this chapter, plain existence of such a transformation is namely just a consequence of the soundness theorem for $\mathbf{HB}^{=}$ with respect to $=_\mu$, and the fact that the completeness theorem for $\mathbf{HB}_{\mathbf{0}}^{=}$ with respect to $=_\mu$ can be "made effective" (in the sense that, for every pair $\langle \tau, \sigma \rangle$ of recursive types that are strongly equivalent, a derivation in $\mathbf{HB}_{\mathbf{0}}^{=}$ without open assumptions and with conclusion $\tau = \sigma$ can effectively be built[3]). But such an argumentation only leads to a transformation that completely ignores its input consisting of a derivation $\mathcal{D}$ in $\mathbf{HB}^{=}$ without open assumptions and that instead builds up a mimicking derivation for $\mathcal{D}$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ *purely from scratch*. This is clearly not what one has in mind as a proof-theoretic transformation. In contrast with such a transformation, here we will develop a sequence of effective, concrete and mainly locally applied manipulation steps that can be applied to an arbitrary given derivation $\mathcal{D}$ in $\mathbf{HB}^{=}$ with the outcome of a mimicking derivation $\mathcal{D}'$ for $\mathcal{D}$ in $\mathbf{HB}_{\mathbf{0}}^{=}$.

Now we give the main theorem of this section, which states the existence of an effective *proof-theoretic transformation* from derivations in the system $\mathbf{HB}^{=}$ without open assumptions into mimicking derivations in the system $\mathbf{HB}_{\mathbf{0}}^{=}$. The proof of this theorem is going to proceed by developing an effective method for 'normalizing' derivations of $\mathbf{HB}_{\mathbf{0}}^{=} + \mathrm{SYMM} + \mathrm{TRANS}$ by "effective SYMM- and TRANS-elimination" (which will be performed by a similar technique as is applied in classical proofs for cut-elimination in Gentzen systems). This method could indeed also be called a 'normalization procedure' for $\mathbf{HB}_{\mathbf{0}}^{=}$-derivations because it enables to transform derivations in $\mathbf{HB}^{=}$, and thus derivations that do not necessarily fulfill the subformula property SP_1, into mimicking derivations in $\mathbf{HB}_{\mathbf{0}}^{=}$, which fulfill SP_1 as a consequence of the fact that $\mathbf{HB}_{\mathbf{0}}^{=}$ obeys the subformula property SP_1.

Theorem 8.2.2. *Every derivation $\mathcal{D}$ in $\mathbf{HB}^{=}$ without open assumptions can be transformed, by an effective proof-theoretic transformation, into a derivation $\mathcal{D}'$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ with the same conclusion and without open assumptions.*

[3] For this, an algorithm for building $\mathbf{HB}_{\mathbf{0}}^{=}$-derivations with equations between strongly equivalent recursive types as conclusions can be used that is similar to Algorithm S given in [BrHe98] on page 11.

The *proof* for this theorem will be given below on page 318. For this proof we will invoke three main lemmas, Lemma 8.2.3, Lemma 8.2.13, and Lemma 8.2.21, which will formalize the three principal steps in our demonstration of Theorem 8.2.2. Before successively stating and proving these lemmas, we give an approximate outline of the steps in the transformation to which these lemmas correspond respectively.

By a slightly simplified account, the three main steps in our proof of Theorem 8.2.2 consist in the proof of the assertions that are formulated and explained in the following items (St1), (St2), and (St3):

(St1) Every derivation in $\mathbf{HB}^{=}$ can be transformed into a derivation in $\mathbf{HB}_{\mathbf{0}}^{=} + {} + \text{SYMM} + \text{TRANS}$ in a very easy and straightforward way.

(St2) For every derivation $\mathcal{D}$ in $\mathbf{HB}_{\mathbf{0}}^{=} + \text{SYMM} + \text{TRANS}$ with conclusion $\tau = \sigma$, for some $\tau, \sigma \in \mu Tp$, and without open assumption classes it is possible, by performing the effective operations of

(a) permuting applications of rules SYMM and TRANS upwards over applications $\mathbf{HB}_{\mathbf{0}}^{=}$-rules, and of

(b) 'unfolding' applications of ARROW/FIX,

to build up, in a stepwise and effective way, a finite or infinite sequence $\mathcal{SD} = {} = \langle \mathcal{D}^{(n)} \rangle_{n \in I}$, where $I = \omega$ or $I = [0, n_{\max}] \cap \omega$ for some $n_{\max} \in \omega$, of derivations in $\mathbf{HB}_{\mathbf{0}}^{=} + \text{SYMM} + \text{TRANS}$ such that

- $\mathcal{SD}$ starts with $\mathcal{D}$, i.e. $\mathcal{D}^{(0)}$ is $\mathcal{D}$,
- for all $n \in I$, the derivation $\mathcal{D}^{(n)}$ has conclusion $\tau = \sigma$ and does not have open assumptions, i.e. $\mathcal{D}^{(n)}$ mimics $\mathcal{D}$,
- either $\mathcal{SD}$ is finite and it ends, for some $n_{\max} \in \omega$, with a derivation $\mathcal{D}^{(n_{\max})}$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ (and hence with a derivation without SYMM- and TRANS-applications),
- or $\mathcal{SD}$ is infinite and the minimal heights h_n of applications of SYMM or TRANS in the derivations $\mathcal{D}^{(n)}$ diverge against infinity, i.e. $\langle h_n \rangle_{n \in \omega} \to \infty$ holds (and hence all derivations in the sequence $\mathcal{SD}$ contain applications of SYMM and/or of TRANS).

The construction of the sequence $\mathcal{SD}$ ensures that, if the applications of SYMM and TRANS are not eliminated during the construction steps of the sequence $\mathcal{SD}$, as result of successively permuting them upwards as far as possible in derivations $\mathcal{D}^{(n)}$ and of repeatedly 'unfolding' these derivations appropriately, then at least the "$\mathbf{HB}_{\mathbf{0}}^{=}$-end-parts" of the derivations $\mathcal{D}^{(n)}$ get larger and larger (in the sense that the heights h_n of the bottommost of applications SYMM or TRANS grow and tend to infinity). Hereby we have called, and will do so again below in (St3), the SYMM- and TRANS-free end-part of a derivation $\tilde{\mathcal{D}}$ also the $\mathbf{HB}_{\mathbf{0}}^{=}$-*end-part* of $\tilde{\mathcal{D}}$; it will also be denoted by $\lceil \tilde{\mathcal{D}} \rceil_{\mathbf{HB}_{\mathbf{0}}^{=}}$ (cf. Definition 8.2.11 for two precise notions that formalize this concept).

(St3) From every infinite sequence $\mathcal{SD} = \langle \mathcal{D}^{(n)} \rangle_{n\in\omega}$ of derivations in the system $\mathbf{HB}_{\mathbf{0}}^{=} + \mathrm{SYMM} + \mathrm{TRANS}$ with conclusion $\tau = \sigma$ and without open assumption classes such that $\langle h_n \rangle_{n\in\omega} \to \infty$ (where the numbers h_n are, as in (St2), the minimal heights in $\mathcal{D}^{(n)}$ of an application of SYMM and TRANS) a derivation $\mathcal{D}'$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ (and hence a derivation without applications of SYMM and TRANS) without open assumptions and with conclusion $\tau = \sigma$ can eventually be extracted; more precisely, for all $n \in \omega$ such that $h_n \geq h_{(\tau,\sigma)}$ in relation to a certain bound $h_{(\tau,\sigma)}$, which depends on the sizes and depths of the recursive types τ and σ in the conclusion $\tau = \sigma$ of $\mathcal{D}^{(n)}$, the $\mathbf{HB}_{\mathbf{0}}^{=}$-end-part $\lceil \mathcal{D}^{(n)} \rceil_{\mathbf{HB}_{\mathbf{0}}^{=}}$ of $\mathcal{D}^{(n)}$ can be effectively be transformed into a derivation $\mathcal{D}'$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ with conclusion $\tau = \sigma$ and *without open assumption classes*. This transformation can then be carried out by dropping unnecessary subderivations and by allowing appropriate back-bindings of then newly arising open assumptions to respective occurrences of applications[4] ARROW or ARROW/FIX present in $\lceil \mathcal{D}^{(n)} \rceil_{\mathbf{HB}_{\mathbf{0}}^{=}}$.

From the assertions in (St1)–(St3) above an effective algorithm for transforming derivations in $\mathbf{HB}^{=}$ without open assumptions into mimicking derivations in $\mathbf{HB}_{\mathbf{0}}^{=}$ can actually be extracted that acts as follows: given a derivation $\mathcal{D}$ in $\mathbf{HB}^{=}$ without open assumption classes and with conclusion $\tau = \sigma$, first perform (St1) with a mimicking derivation $\tilde{\mathcal{D}}$ in $\mathbf{HB}_{\mathbf{0}}^{=} + \mathrm{SYMM} + \mathrm{TRANS}_1$ as the result; then construct stepwisely a sequence $\mathcal{SD} = \langle \mathcal{D}^{(0)}, \mathcal{D}^{(1)}, \mathcal{D}^{(2)}, \ldots \rangle$ of derivations in $\mathbf{HB}_{\mathbf{0}}^{=} + \mathrm{SYMM} +$ $+ \{\mathrm{TRANS}_k\}_k$ that is guaranteed by (St2) and that starts with $\mathcal{D}^{(0)} = \tilde{\mathcal{D}}$; proceed with the construction of this sequence as long as $h_n < h_{(\tau,\sigma)}$ holds for an $n \in \omega$, that is, as long as the minimal height h_n in $\mathcal{D}^{(n)}$ of an application of SYMM or TRANS is less than a certain bound $h_{(\tau,\sigma)}$ given below that depends on τ and σ; and finally, if for the first time $h_n \geq h_{(\tau,\sigma)}$ holds for an $n \in \omega$ (this always happens eventually because $\langle h_n \rangle_n \to_{n\to\infty} \infty$ is guaranteed by the construction of the sequence $\mathcal{SD}$ according to (St2)), extract a $\mathbf{HB}_{\mathbf{0}}^{=}$-derivation $\mathcal{D}'$ without open assumptions and with conclusion $\tau = \sigma$ from the $\mathbf{HB}_{\mathbf{0}}^{=}$-end-derivation $\lceil \mathcal{D}^{(n)} \rceil_{\mathbf{HB}_{\mathbf{0}}^{=}}$ of $\mathcal{D}^{(n)}$ (this is possible due to the assertion in (St3)). We will give a flow-chart like illustration of a refined version of this algorithm in Figure 8.4 below.

The description we have given above of the main steps in the transformation between $\mathbf{HB}^{=}$-derivations without open assumptions and mimicking derivations in $\mathbf{HB}_{\mathbf{0}}^{=}$ was simplified insofar, as it will actually not be TRANS-applications that are permuted upwards in derivations but applications of a variant rule TRANS_1 of TRANS (upwards-permutations of SYMM will however indeed be used); applications of the rule TRANS_1 allow to take variants of the recursive types in the two equations in its premises first before applying the usual transitivity rule. Furthermore, we will also use still more general transitivity rules TRANS_k, for all $k \in \omega\backslash\{0\}$, during steps of the transformation, where an application of TRANS_k can be mimicked by a 'cascade' of k TRANS_1-applications. And lastly, we will also use more general versions $^*(\mu\!-\!\bot)^{\bot\mathrm{der}}_{l/r}$ of the rules $(\mu\!-\!\bot)^{\bot\mathrm{der}}_{l/r}$ of $\mathbf{HB}_{\mathbf{0}}^{=}$. Although

[4] Strictly speaking, a back-binding to an application of ARROW is not possible literally, but in this case a renaming of this application to an application of ARROW/FIX is necessary.

we will utilize these additional rules for intermediary steps of the transformation and although the lemmas in our proof of Theorem 8.2.2 refer to applications of the rules $\{\mathrm{TRANS}_k\}_k$ (one also to applications of ${}^*(\mu-\bot)^{\bot\mathrm{der}}_{l/r}$), the above account in steps (St1)–(St3) could be reconstructed in the form of precise statements. This is due to the fact shown below that all mentioned additional rules are derivable in $\mathbf{HB}_{\mathbf{0}}^{=}$ and that henceforth their applications can be eliminated from derivations in the extension of $\mathbf{HB}_{\mathbf{0}}^{=}$ by adding these rules.

With these additional rules, the steps (St1)′, (St2)′, and (St3)′ in our proof of Theorem 8.2.2 below on page 318 will be the result of letting the role of the rule TRANS be taken over by generalized transitivity rules of the family $\{\mathrm{TRANS}_k\}_k$: step (St1)′ will consist in the proof of the assertion that every derivation in $\mathbf{HB}_{\mathbf{0}}^{=}$ can be transformed in a straightforward way into a mimicking derivation $\tilde{\mathcal{D}}$ in $\mathbf{HB}_{\mathbf{0}}^{=} +$ $+ \mathrm{SYMM} + \{\mathrm{TRANS}_k\}_k$; step (St2)′ will justify the assertion that for every derivation $\mathcal{D}$ in $\mathbf{HB}_{\mathbf{0}}^{=} + \mathrm{SYMM} + \{\mathrm{TRANS}_k\}_k$ a sequence $\mathcal{SD} = \langle \mathcal{D}^{(n)} \rangle_{n\in I}$ of mimicking derivations in $\mathbf{HB}_{\mathbf{0}}^{=} + \mathrm{SYMM} + \{\mathrm{TRANS}_k\}_k$ can be built effectively such that the minimal heights h_n of applications of SYMM or rules from $\{\mathrm{TRANS}_k\}_k$ grow and tend to infinity; and in (St3)′ the assertion will be proved that from every *such* sequence $\mathcal{SD} = \langle \mathcal{D}^{(n)} \rangle_n$ of derivations a derivation $\mathcal{D}'$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ without open assumptions and with the same conclusion as $\mathcal{D}^{(0)}$ can eventually be extracted effectively. Hereby step (St1)′ will be justified by Lemma 8.2.3, step (St2)′ by Lemma 8.2.13, and (St3)′ by Lemma 8.2.21.

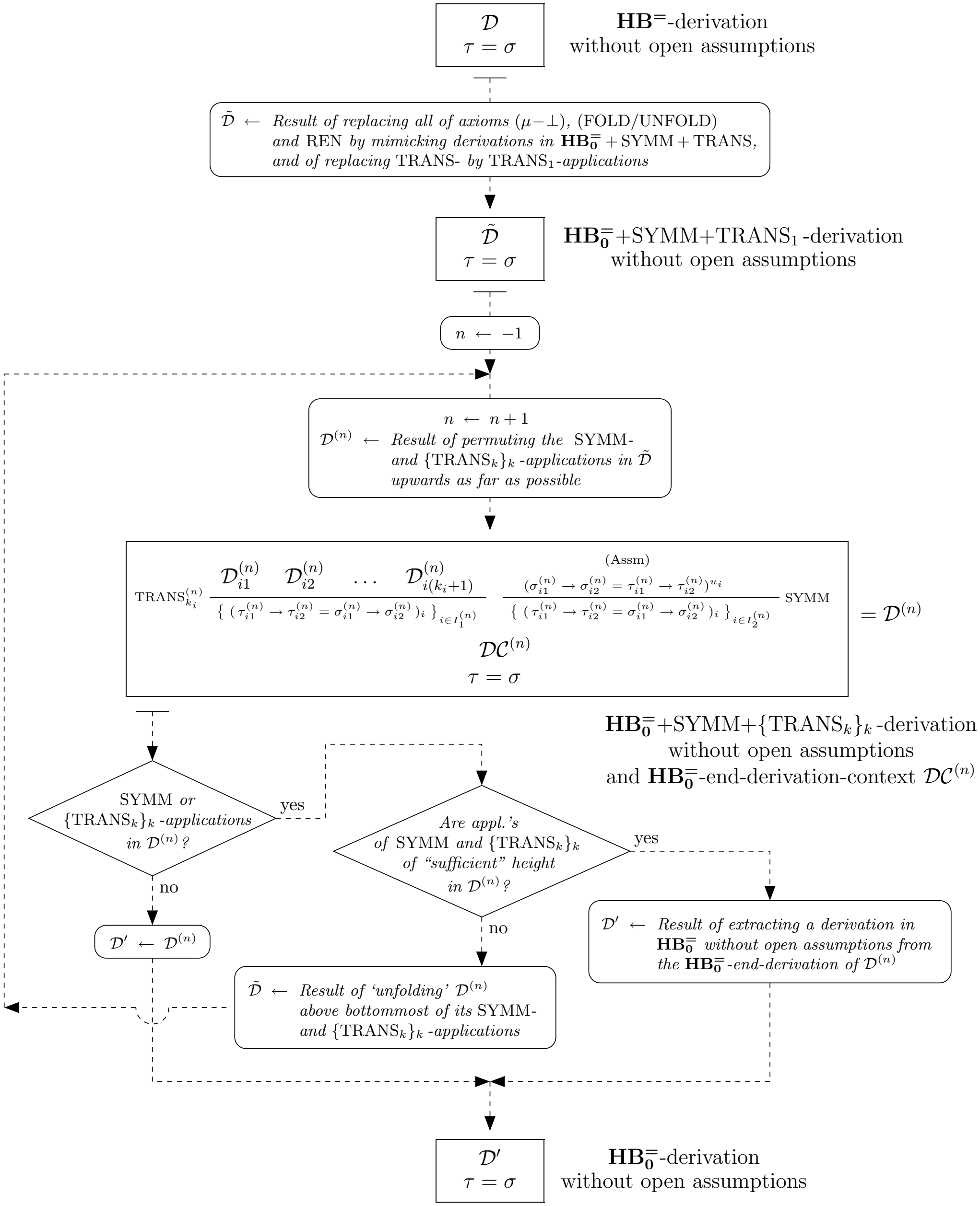
Similarly as described above for the simplified steps (St1)–(St3), also from the actual steps (St1)′, (St2)′, and (St3)′ just outlined of the proof later of Theorem 8.2.2 an effective proof transformation algorithm can be extracted. An illustration of such an algorithm as a flow-chart-like picture is given in Figure 8.4. We place this figure here in the intention of giving some further outline of the transformation developed in this section, notwithstanding the fact that some details appearing in it will only be explained later on.

The actual steps (St1)′–(St3)′ of the eventual proof of Theorem 8.2.2 that have been sketched above correspond roughly to the following action illustrated in the flow-chart in Figure 8.4: step (St1)′ corresponds to the first action taken at the top between the derivations $\mathcal{D}$ and $\tilde{\mathcal{D}}$; the construction in step (St2)′ of a sequence $\mathcal{SD}$ in $\mathbf{HB}_{\mathbf{0}}^{=} + \mathrm{SYMM} + \{\mathrm{TRANS}_k\}_k$ corresponds to repeated executions of the single loop in this flow-chart (and this construction would be continued indefinitely if this loop were never left); and finally step (St3)′ corresponds to the action taken as soon as this loop is left (actually also the second test in this loop, the conditional dependent on whether applications of SYMM and $\{\mathrm{TRANS}_k\}_k$ are of sufficient “height” in $\mathcal{D}^{(n)}$, will be part of step (St3)′ of the proof).

We are going to motivate and introduce the mentioned additional rules and gather some of their later needed, basic properties below after the following lemma that justifies the first step (St1) of the transformation as described in our tentative account given above (and in fact, this lemma will also be the main part of step (St1)′ in the proof of Theorem 8.2.2).

Lemma 8.2.3. *Every derivation $\mathcal{D}$ in $\mathbf{HB}_{\mathbf{0}}^{=}$, with possibly open assumptions, can*

Figure 8.4: Illustration as a flow-chart of the transformation developed in this section from derivations in the Brandt-Henglein system $\mathbf{HB}^{=}$ without open assumptions into mimicking derivations in the variant Brandt-Henglein system $\mathbf{HB}_{\mathbf{0}}^{=}$.


$\mathcal{D}$
$\tau = \sigma$
$\mathbf{HB}^{=}$-derivation without open assumptions
$\tilde{\mathcal{D}} \leftarrow$ *Result of replacing all of axioms* $(\mu-\bot)$, (FOLD/UNFOLD) *and* REN *by mimicking derivations in* $\mathbf{HB}_{\mathbf{0}}^{=}$ + SYMM + TRANS, *and of replacing* TRANS- *by* TRANS_1*-applications*
$\tilde{\mathcal{D}}$
$\tau = \sigma$
$\mathbf{HB}_{\mathbf{0}}^{=}$+SYMM+$\text{TRANS}_1$ -derivation without open assumptions
$n \leftarrow -1$
$n \leftarrow n+1$
$\mathcal{D}^{(n)} \leftarrow$ *Result of permuting the* SYMM- *and* $\{\text{TRANS}_k\}_k$ *-applications in* $\tilde{\mathcal{D}}$ *upwards as far as possible*
$\text{TRANS}_{k_i}^{(n)}$
$\mathcal{D}_{i1}^{(n)} \quad \mathcal{D}_{i2}^{(n)} \quad \dots \quad \mathcal{D}_{i(k_i+1)}^{(n)}$
$\{ \, (\tau_{i1}^{(n)} \to \tau_{i2}^{(n)} = \sigma_{i1}^{(n)} \to \sigma_{i2}^{(n)})_i \, \}_{i \in I_1^{(n)}}$
(Assm)
$(\sigma_{i1}^{(n)} \to \sigma_{i2}^{(n)} = \tau_{i1}^{(n)} \to \tau_{i2}^{(n)})^{u_i}$
$\{ \, (\tau_{i1}^{(n)} \to \tau_{i2}^{(n)} = \sigma_{i1}^{(n)} \to \sigma_{i2}^{(n)})_i \, \}_{i \in I_2^{(n)}}$
SYMM
$\mathcal{DC}^{(n)}$
$\tau = \sigma$
$= \mathcal{D}^{(n)}$
$\mathbf{HB}_{\mathbf{0}}^{=}$+SYMM+$\{\text{TRANS}_k\}_k$ -derivation without open assumptions and $\mathbf{HB}_{\mathbf{0}}^{=}$-end-derivation-context $\mathcal{DC}^{(n)}$
SYMM *or* $\{\text{TRANS}_k\}_k$ *-applications in* $\mathcal{D}^{(n)}$*?*
yes
no
Are appl.'s of SYMM *and* $\{\text{TRANS}_k\}_k$ *of "sufficient" height in* $\mathcal{D}^{(n)}$*?*
yes
no
$\mathcal{D}' \leftarrow \mathcal{D}^{(n)}$
$\tilde{\mathcal{D}} \leftarrow$ *Result of 'unfolding'* $\mathcal{D}^{(n)}$ *above bottommost of its* SYMM- *and* $\{\text{TRANS}_k\}_k$ *-applications*
$\mathcal{D}' \leftarrow$ *Result of extracting a derivation in* $\mathbf{HB}_{\mathbf{0}}^{=}$ *without open assumptions from the* $\mathbf{HB}_{\mathbf{0}}^{=}$*-end-derivation of* $\mathcal{D}^{(n)}$
$\mathcal{D}'$
$\tau = \sigma$
$\mathbf{HB}_{\mathbf{0}}^{=}$-derivation without open assumptions

effectively be transformed into a derivation $\mathcal{D}'$ in $\mathbf{HB}_{\mathbf{0}}^{=} + \mathrm{SYMM} + \mathrm{TRANS}$ *that mimics* $\mathcal{D}$.

Proof. Every derivation $\mathcal{D}$ in $\mathbf{HB}^{=}$ can be transformed into a derivation $\mathcal{D}'$ in $\mathbf{HB}_{\mathbf{0}}^{=} + \mathrm{SYMM} + \mathrm{TRANS}$ with the same conclusion and with the same open assumption classes by performing the following two kinds of actions:

(a) Rename all those applications of ARROW/FIX, at which no assumptions are discharged, into applications of ARROW.

(b) Replace all occurrences of axioms (REN), $(\mu-\bot)'$ and (FOLD/UNFOLD) at the top of the derivation by respective derivations consisting only of an axiom (REFL) that is followed by a single application of REN, $(\mu\text{–}\bot)_l^{\bot\mathrm{der}}$, or FOLD_l, respectively.

□

Upwards-permutation of applications of TRANS over applications of rules belonging to $\mathbf{HB}_{\mathbf{0}}^{=}$ is not an entirely straightforward matter, for at least three reasons. Firstly, it is not obvious how to proceed in the situation of a derivation $\mathcal{D}$ in $\mathbf{HB}_{\mathbf{0}}^{=} + {}$ $+ \mathrm{TRANS}$ like

$$\dfrac{\mathrm{REN}\;\dfrac{\begin{array}{c}\mathcal{D}_{11}\\ \tau' = \rho'\end{array}}{\tau = \rho} \qquad \dfrac{\begin{array}{c}\mathcal{D}_{21}\\ \rho'' = \sigma'\end{array}}{\rho = \sigma}\;\mathrm{REN}}{\tau = \sigma}\;\mathrm{TRANS}$$

where the immediate subderivations of $\mathcal{D}$ both end with applications of REN. Secondly, also in the situation of a derivation $\mathcal{D}$ in $\mathbf{HB}_{\mathbf{0}}^{=} + \mathrm{TRANS}$ of the form

$$\dfrac{(\mu-\bot)_r^{\bot\mathrm{der}}\;\dfrac{\begin{array}{c}\mathcal{D}_{11}\\ \tau = \bot\end{array}}{\tau = \mu\alpha_1\alpha_2.\,\alpha_1} \qquad \dfrac{\begin{array}{c}\mathcal{D}_{21}\\ \mu\alpha_1\alpha_2\alpha_1.\,\alpha_1 = \sigma\end{array}}{\mu\alpha_1\alpha_2.\,\alpha_1 = \sigma}\;\mathrm{FOLD}_l}{\tau = \sigma}\;\mathrm{TRANS}$$

the bottommost application of TRANS cannot directly be permuted upwards over one or over both of the last rule applications in immediate subderivations. And thirdly, a TRANS-application that follows upon an application of ARROW/FIX in, say, the right premise and an application of ARROW or ARROW/FIX in the left premise cannot be permuted upwards in a straightforward manner; furthermore also the case with a marked assumption in a premise of a TRANS-application poses a similar difficulty. For overcoming the first and second problems, we will respectively introduce, for auxiliary purposes, variant rules TRANS_1 of TRANS and ${}^*(\mu-\bot)_{l/r}^{\bot\mathrm{der}}$ of $(\mu-\bot)_{l/r}^{\bot\mathrm{der}}$. And the third problem will later be dealt with by introducing operations that can be used to 'unfold' derivations in $\mathbf{HB}_{\mathbf{0}}^{=} + \mathrm{SYMM} + {}$ $+ \{\mathrm{TRANS}_k\}_k$ above marked assumptions or above conclusions of ARROW/FIX-applications (where the family $\{\mathrm{TRANS}_k\}_k$ of generalized transitivity rules will also be defined below).

Of the mentioned additional rules, we first introduce TRANS_1 and ${}^*(\mu-\bot)_{l/r}^{\bot\mathrm{der}}$.

Definition 8.2.4 (The rules $\mathbf{TRANS_1}$ and $^{*}(\mu\text{–}\bot)^{\bot\mathbf{der}}_{l/r}$). In the items (i) and (ii) below we define the rules TRANS_1, and the rules $^{*}(\mu\text{–}\bot)^{\bot\mathrm{der}}_{l}$ as well as $^{*}(\mu\text{–}\bot)^{\bot\mathrm{der}}_{r}$ by stipulating, for an arbitrary natural-deduction system $\mathcal{S}$ with $\mu Tp\text{–}Eq$ as its set of formulas, what applications these rules respectively enable when added to $\mathcal{S}$.

(i) Applications of TRANS_1 at the bottom of derivations $\mathcal{D}$ in $\mathcal{S}+\mathrm{TRANS}_1$ have the form

$$\dfrac{\begin{array}{c}\mathcal{D}_1\\ \tau' = \rho\end{array}\qquad\begin{array}{c}\mathcal{D}_2\\ \rho' = \sigma'\end{array}}{\tau = \sigma}\ \mathrm{TRANS}_1\quad \text{(given that (*))} \tag{8.24}$$

where $\tau, \tau', \rho, \rho', \sigma, \sigma' \in \mu Tp$ and where the condition (*) means that $\tau \equiv_{\mathrm{ren}} \tau'$, $\rho \equiv_{\mathrm{ren}} \rho'$ and $\sigma \equiv_{\mathrm{ren}} \sigma'$.

(ii) Applications of $^{*}(\mu\text{–}\bot)^{\bot\mathrm{der}}_{l}$ and $^{*}(\mu\text{–}\bot)^{\bot\mathrm{der}}_{r}$ at the bottom of derivations $\mathcal{D}$ in $\mathcal{S}+^{*}(\mu\text{–}\bot)^{\bot\mathrm{der}}_{l/r}$ have the respective forms

$$^{*}(\mu\text{–}\bot)^{\bot\mathrm{der}}_{l}\ \dfrac{\begin{array}{c}\mathcal{D}_1\\ \bot = \sigma\end{array}}{\mu\alpha_1\alpha_2\ldots\alpha_n.\,\alpha_i = \sigma}\qquad\qquad \dfrac{\begin{array}{c}\mathcal{D}_1\\ \tau = \bot\end{array}}{\tau = \mu\beta_1\beta_2\ldots\beta_n.\,\beta_i}\ ^{*}(\mu\text{–}\bot)^{\bot\mathrm{der}}_{r} \tag{8.25}$$

where $n \in \omega\backslash\{0\}$, $\alpha_1, \ldots, \alpha_n, \beta_1, \ldots, \beta_n \in TVar$, $i \in \{1, \ldots, n\}$, and $\tau, \sigma \in$ $\in \mu Tp$.

☒

The following proposition formulates the obvious observation that every application of TRANS_1 can be mimicked using an application of TRANS with an application of REN above each of its premises.

Proposition 8.2.5. *Let $\mathcal{S}$ be a natural-deduction system with $\mu Tp\text{–}Eq$ as its set of formulas and with the property that $\mathcal{S}$ contains the rules* TRANS *and* REN*. Then the rule* TRANS_1 *is derivable in $\mathcal{S}$. In particular,* TRANS_1 *is derivable in* $\mathbf{HB}^{=}_{\mathbf{0}}$*.*

Also applications of the rules $^{*}(\mu-\bot)^{\bot\mathrm{der}}_{l/r}$ can be mimicked with the help of $\mathbf{HB}^{=}_{\mathbf{0}}$-rules. More precisely, every application of $^{*}(\mu-\bot)^{\bot\mathrm{der}}_{l}$ or $^{*}(\mu-\bot)^{\bot\mathrm{der}}_{r}$ of respective form in (8.25), with $n \in \omega\backslash\{0\}$ and $i \in \{1, \ldots, n\}$, and with $\alpha_1, \ldots, \alpha_n \in$ $\in TVar$, $\sigma \in \mu Tp$, or respectively with $\beta_1, \ldots, \beta_n \in TVar$ and $\tau \in \mu Tp$, can be mimicked by a unary derivation-context in $\mathbf{HB}^{=}_{\mathbf{0}}$ without open assumptions of the respective form

$$\begin{array}{rc}(\mu\text{–}\bot)^{\bot\mathrm{der}}_{l} & \dfrac{[\,\bot = \sigma\,]_1}{\mu\alpha_{j_1}\ldots\alpha_n.\,\alpha_i = \sigma}\\ \mathrm{FOLD}_l & \overline{\overline{\mu\alpha_1\ldots\alpha_n.\,\alpha_i = \sigma}}\end{array}\quad\text{or}\quad\begin{array}{cl}\dfrac{[\,\tau = \bot\,]_1}{\tau = \mu\beta_{j_2}\ldots\beta_n.\,\beta_i} & (\mu\text{–}\bot)^{\bot\mathrm{der}}_{r}\\ \overline{\overline{\tau = \mu\beta_1\ldots\beta_n.\,\beta_i}} & \mathrm{FOLD}_r\end{array} \tag{8.26}$$

(for reading convenience, we have inserted the premises of the mimicked applications into the context-holes $[\,]_1$ of these derivation-contexts), where $j_1 \in \{1,\ldots,n\}$ on the left-hand side is such that $\alpha_1,\ldots,\alpha_{j_1-1} \not\equiv \alpha_i$ and $\alpha_{j_1} \equiv \alpha_i$, and where $j_2 \in \{1,\ldots,n\}$ on the right-hand side is such that $\beta_1,\ldots,\beta_{j_2-1} \not\equiv \beta_i$ and $\beta_{j_2} \equiv \beta_i$ (these particular stipulations prevent nlμb-decreasing applications of $\mathrm{FOLD}_{l/r}$ from occurring in derivation-contexts of the form (8.26))[5].

By a $({}^{*}(\mu{-}\bot)^{\bot\mathrm{der}}_{l/r} \rightsquigarrow (\mu{-}\bot)^{\bot\mathrm{der}}_{l/r},\ \mathrm{FOLD}_{l/r})$*-mimicking step* we will mean an arbitrary replacement step within a derivation $\mathcal{D}$ in $\mathbf{HB}_{\mathbf{0}}^{=} + \mathrm{SYMM} + \{\mathrm{TRANS}_k\}_k + {}+{}^{*}(\mu-\bot)^{\bot\mathrm{der}}_{l/r}$ of an application of ${}^{*}(\mu-\bot)^{\bot\mathrm{der}}_{l}$ or ${}^{*}(\mu-\bot)^{\bot\mathrm{der}}_{r}$ by a respective mimicking derivation-context as displayed in (8.26). The possibility of performing such mimicking steps demonstrates the assertion of the following proposition.

Proposition 8.2.6. *Let $\mathcal{S}$ be a natural-deduction system such that μTp–Eq is its set of formulas and such that $\mathcal{S}$ contains the rules $(\mu{-}\bot)^{\bot\mathrm{der}}_{l/r}$ and* $\mathrm{FOLD}_{l/r}$. *Then the rules ${}^{*}(\mu{-}\bot)^{\bot\mathrm{der}}_{l/r}$ are derivable in $\mathcal{S}$. In particular, the rules ${}^{*}(\mu{-}\bot)^{\bot\mathrm{der}}_{l/r}$ are derivable in $\mathbf{HB}_{\mathbf{0}}^{=}$.*

With these new rules it is now possible to show the following lemma, which is going to play a key role in the construction of the second step of the transformation developed in this section. It states conditions under which an application of TRANS_1 is upwards-permutable in derivations in the extension of $\mathbf{HB}_{\mathbf{0}}^{=}$ with the rules SYMM, TRANS_1, and $(\mu{-}\bot)^{\bot\mathrm{der}}_{l/r}$, and it describes the possible outcomes of single upwards-permutation steps of TRANS_1-applications.

Lemma 8.2.7 (Upwards-permuting of $\mathbf{TRANS}_1$-applications). *Let $\mathcal{D}$ be a derivation in $\mathbf{HB}_{\mathbf{0}}^{=} + \mathrm{SYMM} + \mathrm{TRANS}_1 + {}^{*}(\mu{-}\bot)^{\bot der}_{l/r}$, with possibly open assumption classes, that is of the form*

$$\dfrac{\begin{matrix}\mathcal{D}_1 \\ \tau' = \rho\end{matrix} \qquad \begin{matrix}\mathcal{D}_2 \\ \rho' = \sigma'\end{matrix}}{\tau = \sigma}\,\mathrm{TRANS}_1 \tag{8.27}$$

such that the subderivations $\mathcal{D}_1$ and $\mathcal{D}_2$ are of respective depth $|\mathcal{D}_1|, |\mathcal{D}_2| \geq 1$, and $\mathcal{D}_1$ and $\mathcal{D}_2$ end with applications of rules of $\mathbf{HB}_{\mathbf{0}}^{=} + {}^{*}(\mu{-}\bot)^{\bot der}_{l/r}$ other than ARROW/FIX.

Then $\mathcal{D}$ can be transformed, by permuting the application of TRANS_1 *at the bottom of $\mathcal{D}$ upwards over one or over both of the last rule applications in $\mathcal{D}_1$ and $\mathcal{D}_2$, into a derivation $\tilde{\mathcal{D}}$ in $\mathbf{HB}_{\mathbf{0}}^{=} + \mathrm{SYMM} + \mathrm{TRANS}_1 + {}^{*}(\mu{-}\bot)^{\bot der}_{l/r}$ of one of the following three forms:*

$$\dfrac{\dfrac{\dfrac{\tilde{\mathcal{D}}_{111} \qquad \tilde{\mathcal{D}}_{112}}{\tilde{\tau}_1 = \tilde{\sigma}_1}\,\mathrm{TRANS}_1}{\tilde{\tau} = \tilde{\sigma}}\,R}{\tau = \sigma}\,\mathrm{REN} \qquad \textit{or} \qquad \dfrac{\tilde{\mathcal{D}}_1 \qquad \tilde{\mathcal{D}}_2}{\tau = \sigma}\,\mathrm{TRANS}_1\,, \tag{8.28}$$

[5] Our proofs below do not make use of this convenient property of the mimicking derivation-contexts defined here for applications of ${}^{*}(\mu{-}\bot)^{\bot\mathrm{der}}_{l/r}$ because in the transformation we develop applications of nlμb-decreasing applications of $\mathrm{FOLD}_{l/r}$ are repeatedly eliminated *after* the elimination of applications of ${}^{*}(\mu{-}\bot)^{\bot\mathrm{der}}_{l/r}$.

where $|\tilde{\mathcal{D}}_{111}| + |\tilde{\mathcal{D}}_{112}| < |\mathcal{D}_1| + |\mathcal{D}_2|$ *and "R" indicates an application of a rule* $\mathrm{FOLD}_{l/r}$ *or* ${}^*(\mu{-}\bot)^{\bot der}_{l/r}$*, and where respectively* $|\tilde{\mathcal{D}}_1| + |\tilde{\mathcal{D}}_2| < |\mathcal{D}_1| + |\mathcal{D}_2|$ *holds, or*

$$\dfrac{\dfrac{\tilde{\mathcal{D}}_{11} \qquad \tilde{\mathcal{D}}_{12}}{\tau_1 = \sigma_1}\,\mathrm{TRANS}_1 \qquad \dfrac{\tilde{\mathcal{D}}_{21} \qquad \tilde{\mathcal{D}}_{22}}{\tau_2 = \sigma_2}\,\mathrm{TRANS}_1}{\tau = \sigma}\,\mathrm{ARROW} \tag{8.29}$$

where, for each $j \in \{1,2\}$*,* $|\tilde{\mathcal{D}}_{j1}| + |\tilde{\mathcal{D}}_{j2}| < |\mathcal{D}_1| + |\mathcal{D}_2|$ *holds.*

Since upwards-permutation steps of TRANS_1-applications constitute the most important basic operations within our transformation from $\mathbf{HB}^{=}$ to $\mathbf{HB}^{=}_{\mathbf{0}}$, we set out to give the proof for this lemma in some detail.

Proof. Let $\mathcal{D}$ be an arbitrary derivation in the system $\mathbf{HB}^{=}_{\mathbf{0}} + \mathrm{SYMM} + \mathrm{TRANS}_1 + {}+ {}^*(\mu{-}\bot)^{\bot\mathrm{der}}_{l/r}$, with possibly open assumption classes, that is of the form (8.27), with some $\tau, \rho, \sigma \in \mu Tp$ and respective variants $\tau', \rho', \sigma' \in \mu Tp$ of τ, ρ and σ, such that the subderivations $\mathcal{D}_1$ and $\mathcal{D}_2$ have depths $|\mathcal{D}_1|, |\mathcal{D}_2| \geq 1$, and such that $\mathcal{D}_1$ and $\mathcal{D}_2$ end with applications of rules of $\mathbf{HB}^{=}_{\mathbf{0}} + {}^*(\mu{-}\bot)^{\bot\mathrm{der}}_{l/r}$ other than ARROW/FIX.

By the assumptions on $\mathcal{D}$, the last rule applications in $\mathcal{D}_1$ and $\mathcal{D}_2$ must be applications of REN, $(\mu{-}\bot)^{\bot\mathrm{der}}_{l/r}$, $\mathrm{FOLD}_{l/r}$, ARROW or ${}^*(\mu{-}\bot)^{\bot\mathrm{der}}_{l/r}$. Due to this and the fact that the rules $(\mu{-}\bot)^{\bot\mathrm{der}}_{l/r}$ may be looked upon as special cases of the rules ${}^*(\mu-\bot)^{\bot\mathrm{der}}_{l/r}$, it suffices to consider the following six cases for showing the assertion of the lemma:

(1) at least one of $\mathcal{D}_1$ or $\mathcal{D}_2$ ends with an application of REN,

(2) $\mathcal{D}_1$ ends with an application of FOLD_l or ${}^*(\mu{-}\bot)^{\bot\mathrm{der}}_{l}$, or $\mathcal{D}_2$ ends with an application of FOLD_r or ${}^*(\mu{-}\bot)^{\bot\mathrm{der}}_{l}$,

(3) $\mathcal{D}_1$ ends with an application of FOLD_r and $\mathcal{D}_2$ ends with an application of FOLD_l,

(4) $\mathcal{D}_1$ ends with an application of ${}^*(\mu{-}\bot)^{\bot\mathrm{der}}_{r}$ and $\mathcal{D}_2$ ends with an application of ${}^*(\mu{-}\bot)^{\bot\mathrm{der}}_{l}$,

(5) $\mathcal{D}_1$ ends with an application of FOLD_r and $\mathcal{D}_2$ ends with an application of $(\mu{-}\bot)^{\bot\mathrm{der}}_{l}$, or $\mathcal{D}_1$ ends with an application of $(\mu{-}\bot)^{\bot\mathrm{der}}_{r}$ and $\mathcal{D}_2$ ends with an application of FOLD_l,

(6) $\mathcal{D}_1$ and $\mathcal{D}_2$ both end with an application of ARROW.

Apart from (6) and (1) there are no other cases in which an application of ARROW is able to occur at the bottom of $\mathcal{D}_1$ or $\mathcal{D}_2$: it is easy to see that if one of $\mathcal{D}_1$ or $\mathcal{D}_2$ terminates with an application of ARROW, then the other derivation cannot end with an application of $\mathrm{FOLD}_{l/r}$, $(\mu{-}\bot)^{\bot\mathrm{der}}_{l/r}$ or ${}^*(\mu{-}\bot)^{\bot\mathrm{der}}_{l/r}$.

In case (1) the application of REN at the bottom of $\mathcal{D}_1$ or $\mathcal{D}_2$ can simply be "amalgamated" with the application of TRANS_1 at the bottom of $\mathcal{D}$ with a derivation $\tilde{\mathcal{D}}$ as the result that is of the form of the prooftree on the right in (8.28) and that has the desired property.

For settling case (2), we only consider the case with an application of FOLD_l at the bottom of $\mathcal{D}_1$; all other cases can be treated similarly. Therefore we consider the case that $\mathcal{D}$ is of the form

$$\dfrac{\mathrm{FOLD}_l\,\dfrac{\begin{matrix}\mathcal{D}_{11}\\ \tilde\tau_0[\mu\tilde\alpha.\,\tilde\tau_0/\tilde\alpha]=\rho\end{matrix}}{\mu\tilde\alpha.\,\tilde\tau_0=\rho} \qquad \begin{matrix}\mathcal{D}_2\\ \rho'=\sigma'\end{matrix}}{\mu\alpha.\,\tau_0=\sigma}\,\mathrm{TRANS}_1$$

for some $\alpha, \tilde\alpha \in \mathit{TVar}$ and $\tau_0, \tilde\tau_0 \in \mu\mathit{Tp}$ such that $\tau \equiv \mu\alpha.\,\tau_0$ and $\tau' \equiv \mu\tilde\alpha.\,\tilde\tau_0$; from this and from $\tau \equiv_{\mathrm{ren}} \tau'$, a consequence of $\mathcal{D}$ being of the form (8.27), $\mu\alpha.\,\tau_0 \equiv_{\mathrm{ren}}$ $\equiv_{\mathrm{ren}} \mu\tilde\alpha.\,\tilde\tau_0$ follows. In this situation $\mathcal{D}$ can be transformed into the derivation $\tilde{\mathcal{D}}$ of the form

$$\dfrac{\dfrac{\dfrac{\begin{matrix}\mathcal{D}_{11}\\ \tilde\tau_0[\mu\tilde\alpha.\,\tilde\tau_0/\tilde\alpha]=\rho\end{matrix} \qquad \begin{matrix}\mathcal{D}_2\\ \rho'=\sigma'\end{matrix}}{\tilde\tau_0[\mu\tilde\alpha.\,\tilde\tau_0/\tilde\alpha]=\sigma}\,\mathrm{TRANS}_1}{\mu\tilde\alpha.\,\tilde\tau_0=\sigma}\,\mathrm{FOLD}_l}{\mu\alpha.\,\tau_0=\sigma}\,\mathrm{REN}$$

which is of the form of the left prooftree in (8.28) and has the respective desired property.

In case (3) $\mathcal{D}$ is of the form

$$\dfrac{\mathrm{FOLD}_r\,\dfrac{\begin{matrix}\mathcal{D}_{11}\\ \tau'=\rho_0[\mu\alpha.\,\rho_0/\alpha]\end{matrix}}{\tau'=\mu\alpha.\,\rho_0} \qquad \dfrac{\begin{matrix}\mathcal{D}_{21}\\ \tilde\rho_0[\mu\tilde\alpha.\,\tilde\rho_0/\tilde\alpha]=\sigma'\end{matrix}}{\mu\tilde\alpha.\,\tilde\rho_0=\sigma'}\,\mathrm{FOLD}_l}{\tau=\sigma}\,\mathrm{TRANS}_1$$

for some $\alpha, \tilde\alpha \in \mathit{TVar}$ such that $\rho \equiv \mu\alpha.\,\rho_0$ and $\rho' \equiv \mu\tilde\alpha_0.\,\tilde\rho_0$ (which due to $\rho \equiv_{\mathrm{ren}} \rho'$ entails $\mu\alpha.\,\rho_0 \equiv_{\mathrm{ren}} \mu\tilde\alpha.\,\tilde\rho_0$). Here $\mathcal{D}$ can be transformed into the derivation $\tilde{\mathcal{D}}$ of the form

$$\dfrac{\begin{matrix}\mathcal{D}_{11}\\ \tau'=\rho_0[\mu\alpha.\,\rho_0/\alpha]\end{matrix} \qquad \begin{matrix}\mathcal{D}_{21}\\ \tilde\rho_0[\mu\tilde\alpha.\,\tilde\rho_0/\tilde\alpha]=\sigma'\end{matrix}}{\tau=\sigma}\,\mathrm{TRANS}_1$$

due to Lemma 3.4.2, (3.24). The transformed derivation $\tilde{\mathcal{D}}$ is of the form of the prooftree on the right in (8.28) and has the desired property.

As an example for case (5), we consider a situation in which $\mathcal{D}_1$ ends with an application of ${}^*(\mu-\bot)^{\bot\mathrm{der}}_{l/r}$ and $\mathcal{D}_2$ ends with an application of $\mathrm{FOLD}_{l/r}$, more precisely, we consider a situation in which the bottommost application in $\mathcal{D}_1$ corresponds to an application of $(\mu-\bot)^{\bot\mathrm{der}}_{r}$; it is easy to check that all other subcases of this case can be settled similarly. We assume that $\mathcal{D}$ is of the form

$$\dfrac{{}^{*}(\mu\text{–}\bot)^{\bot\text{der}}_{r}\,\dfrac{\begin{matrix}\mathcal{D}_{11}\\ \tau'=\bot\end{matrix}}{\tau'=\mu\alpha_1\ldots\alpha_n.\,\alpha_1} \qquad \dfrac{\begin{matrix}\mathcal{D}_{21}\\ \mu\tilde{\alpha}_2\ldots\tilde{\alpha}_n\tilde{\alpha}_1\ldots\tilde{\alpha}_n.\,\tilde{\alpha}_1=\sigma'\end{matrix}}{\mu\tilde{\alpha}_1\ldots\tilde{\alpha}_n.\,\tilde{\alpha}_1=\sigma'}\,\mathrm{FOLD}_l}{\tau=\sigma}\,\mathrm{TRANS}_1$$

for some $n\in\omega\backslash\{0\}$, $\alpha_1,\ldots,\alpha_n,\tilde{\alpha}_1,\ldots,\tilde{\alpha}_n\in\mu Tp$ such that $\mu\tilde{\alpha}_1\ldots\tilde{\alpha}_n.\,\tilde{\alpha}_1$ is a variant of $\mu\alpha_1\ldots\alpha_n.\,\alpha_1$. Here $\mathcal{D}$ can be transformed into the derivation $\tilde{\mathcal{D}}$ of the form

$$\dfrac{{}^{*}(\mu\text{–}\bot)^{\bot\text{der}}_{r}\,\dfrac{\begin{matrix}\mathcal{D}_{11}\\ \tau'=\bot\end{matrix}}{\tau'=\mu\tilde{\alpha}_2\ldots\tilde{\alpha}_n\tilde{\alpha}_1\ldots\tilde{\alpha}_n.\,\tilde{\alpha}_1} \qquad \begin{matrix}\mathcal{D}_{21}\\ \mu\tilde{\alpha}_2\ldots\tilde{\alpha}_n\tilde{\alpha}_1\ldots\tilde{\alpha}_n.\,\tilde{\alpha}_1=\sigma'\end{matrix}}{\tau=\sigma}\,\mathrm{TRANS}_1$$

which is of the form of the prooftree on the right in (8.28) and which has the respective desired property.

In case (6) the derivation $\mathcal{D}$ is of the form

$$\dfrac{\mathrm{ARROW}\,\dfrac{\begin{matrix}\mathcal{D}_{11}\\ \tau_1'=\rho_1\end{matrix} \qquad \begin{matrix}\mathcal{D}_{12}\\ \tau_2'=\rho_2\end{matrix}}{\tau_1'\to\tau_2'=\rho_1\to\rho_2} \qquad \dfrac{\begin{matrix}\mathcal{D}_{21}\\ \rho_1'=\sigma_1'\end{matrix} \qquad \begin{matrix}\mathcal{D}_{22}\\ \rho_2'=\sigma_2'\end{matrix}}{\rho_1'\to\rho_2'=\sigma_1'\to\sigma_2'}\,\mathrm{ARROW}}{\tau_1\to\tau_2=\sigma_1\to\sigma_2}\,\mathrm{TRANS}_1$$

for some $\tau_1,\tau_2,\tau_1',\tau_2',\ldots,\sigma_1',\sigma_2'\in\mu Tp$ such that, for each $i\in\{1,2\}$, $\sigma_i'\equiv_{\text{ren}}\sigma_i$ holds as well as $\rho_i'\equiv_{\text{ren}}\rho_i$ and and $\tau_i'\equiv_{\text{ren}}\tau_i$. Here $\mathcal{D}$ can be transformed into the derivation $\tilde{\mathcal{D}}$ of the form

$$\dfrac{\mathrm{TRANS}_1\,\dfrac{\begin{matrix}\mathcal{D}_{11}\\ \tau_1'=\rho_1\end{matrix} \qquad \begin{matrix}\mathcal{D}_{21}\\ \rho_1'=\sigma_1'\end{matrix}}{\tau_1=\sigma_1} \qquad \dfrac{\begin{matrix}\mathcal{D}_{12}\\ \tau_2'=\rho_2\end{matrix} \qquad \begin{matrix}\mathcal{D}_{22}\\ \rho_2'=\sigma_2'\end{matrix}}{\tau_2=\sigma_2}\,\mathrm{TRANS}_1}{\tau_1\to\tau_2=\sigma_1\to\sigma_2}\,\mathrm{ARROW}$$

which is of the form of the prooftree in (8.29) and has the desired properties.

In this way we have shown the lemma by having settled all six possible cases for a derivation $\mathcal{D}$ that was assumed to be arbitrary, but in accordance with the hypothesis of the lemma.

□

As a consequence of the third kind of problems mentioned above concerning upwards-permutation of the rule TRANS over $\mathbf{HB}_{\mathbf{0}}^{=}$-rules, even topmost applications of transitivity rules TRANS_1 in $\mathbf{HB}_{\mathbf{0}}^{=}+\mathrm{SYMM}+\mathrm{TRANS}_1$ cannot always be removed eventually by permuting them upwards sufficiently often until axioms are reached. The reason is that upwards-permutation movements of TRANS_1-applications may be blocked by the occurrence of marked assumptions in a premise or of an immediate subderivation that ends with an application of ARROW/FIX (note that, in particular, these cases are not covered by Lemma 8.2.7). And furthermore, if non-topmost applications of TRANS_1 are permuted upwards, then such a movement may also be blocked by the appearance of another TRANS_1-application above one of the premises. For dealing with the last mentioned situation, we introduce now more general variants TRANS_k, for all $k \in \omega \backslash \{0\}$, of the rule TRANS_1, applications of which correspond to k TRANS_1-applications. We will see shortly that two TRANS_1-applications in a derivation in $\mathbf{HB}_{\mathbf{0}}^{=} + \mathrm{SYMM} + \mathrm{TRANS}_1$, where one of these applications occurs immediately above the other, can be "amalgamated" into a single application of TRANS_2.

Definition 8.2.8 (The family $\{\mathbf{TRANS}_k\}_k$ of generalized transitivity rules). We define the rules of the family $\{\mathrm{TRANS}_k\}_{k\in\omega\backslash\{0\}}$, which we henceforth write in the abbreviated form $\{\mathrm{TRANS}_k\}_k$, by stipulating for an arbitrary natural-deduction system $\mathcal{S}$ with μTp–Eq as its set of formulas and for arbitrary $k \in \omega\backslash\{0\}$: An application of the rule TRANS_k at the bottom of a derivation $\mathcal{D}$ in $\mathcal{S}$+TRANS_k has the form

$$\frac{\begin{array}{ccccc}\mathcal{D}_1 & \mathcal{D}_2 & & \mathcal{D}_k & \mathcal{D}_{k+1}\\ \tau = \rho_1 & \rho'_1 = \rho_2 & \ldots & \rho'_{k-1} = \rho_k & \rho'_k = \sigma'\end{array}}{\tau = \sigma}\ \mathrm{TRANS}_k\ \text{(given that (**))} \tag{8.30}$$

where $\tau, \tau', \rho_1, \rho'_1, \rho_2, \rho'_2, \ldots, \rho_{k-1}, \rho'_{k-1}, \rho_k, \rho'_k \in \mu Tp$, and where the condition (**) means that $\tau \equiv_{\mathrm{ren}} \tau'$, $\rho_1 \equiv_{\mathrm{ren}} \rho'_1$, $\rho_2 \equiv_{\mathrm{ren}} \rho'_2$, $\ldots$, $\rho_{k-1} \equiv_{\mathrm{ren}} \rho'_{k-1}$, $\rho_k \equiv_{\mathrm{ren}} \rho'_k$, and $\sigma \equiv_{\mathrm{ren}} \sigma'$.

⊠

We have not taken up a rule TRANS_0 into the family $\{\mathrm{TRANS}_k\}_{k\in\omega\backslash\{0\}}$ introduced in this definition because in the special case $k = 0$ the definition of a rule TRANS_k would just coincide with the definition of the rule REN, which is certainly not a transitivity rule.

Applications of rules of $\{\mathrm{TRANS}_k\}_k$ can easily be replaced by TRANS_1-applications: every application of TRANS_k, for some $k \in \omega\backslash\{0\}$, at the bottom of a derivation $\mathcal{D}$ of the form (8.30) can be eliminated by replacing it with k TRANS_1-applications that are, for example, arranged in a cascaded form at the bottom of the derivation $\mathcal{D}'$ of the form

$$\mathrm{TRANS}_1\ \frac{\begin{array}{cc}\mathcal{D}_1 & \mathcal{D}_2\\ \tau' = \rho_1 & \rho'_1 = \rho_2\end{array}}{\tau = \rho_2}$$

$$\mathrm{TRANS}_1\ \frac{\mathrm{TRANS}_1\ \dfrac{\begin{array}{cc}\ddots & \mathcal{D}_k\\ \tau = \rho_{k-1} & \rho'_{k-1} = \rho_k\end{array}}{\tau = \rho_k} \qquad \begin{array}{c}\mathcal{D}_{k+1}\\ \rho'_k = \sigma'\end{array}}{\tau = \sigma}$$

which mimics $\mathcal{D}$. We call such an elimination step of a TRANS_k-application, also if it is performed in a derivation in $\mathbf{HB}^{=}_{\mathbf{0}} + \mathrm{SYMM} + \{\mathrm{TRANS}_k\}_k$ to some subderivation, a *($\{\mathrm{TRANS}_k\}_k \rightsquigarrow \mathrm{TRANS}_1$)-mimicking step.*

The following proposition is an immediate consequence of Proposition 8.2.5 and of the existence of mimicking steps as just described.

Proposition 8.2.9. *Let $\mathcal{S}$ be a natural-deduction system with μTp–Eq as its set of formulas and with the property that $\mathcal{S}$ contains the rules* TRANS *and* REN*. Then, for all $k \in \omega\backslash\{0\}$, the rule TRANS_k is derivable in $\mathcal{S}$. And in particular, all rules of the family $\{\mathrm{TRANS}_k\}_k$ are derivable in $\mathbf{HB}^{=}_{\mathbf{0}}$.*

The principal reason why we have introduced the family of rules $\{\mathrm{TRANS}_k\}_k$ consists in the fact already hinted above that, for all $k_1, k_2 \in \omega\backslash\{0\}$, an application of TRANS_{k_1} can be *amalgamated* with an application of TRANS_{k_2} into an application of $\mathrm{TRANS}_{k_1+k_2}$ if one of these applications occurs immediately above the other. So, for instance, the two displayed applications of TRANS_2 in a derivation $\mathcal{D}$ of the form

$$\dfrac{\begin{array}{c}\mathcal{D}_1\\ \tau = \rho_1\end{array} \qquad \begin{array}{c}\mathcal{D}_2\\ \rho_1' = \rho_2\end{array} \qquad \dfrac{\begin{array}{c}\mathcal{D}_{31}\\ \rho_2'' = \rho_{31}\end{array} \qquad \begin{array}{c}\mathcal{D}_{32}\\ \rho_{31}' = \rho_{32}\end{array} \qquad \begin{array}{c}\mathcal{D}_{33}\\ \rho_{32}' = \sigma''\end{array}}{\rho_{32}' = \sigma''}\,\mathrm{TRANS}_2}{\tau = \sigma}\,\mathrm{TRANS}_2 \tag{8.31}$$

can be amalgamated into the application of TRANS_4 at the bottom of the derivation $\mathcal{D}'$ of the form

$$\dfrac{\begin{array}{c}\mathcal{D}_1\\ \tau = \rho_1\end{array} \qquad \begin{array}{c}\mathcal{D}_2\\ \rho_1' = \rho_2\end{array} \qquad \begin{array}{c}\mathcal{D}_{31}\\ \rho_2'' = \rho_{31}\end{array} \qquad \begin{array}{c}\mathcal{D}_{32}\\ \rho_{31}' = \rho_{32}\end{array} \qquad \begin{array}{c}\mathcal{D}_{33}\\ \rho_{32}' = \sigma''\end{array}}{\tau = \sigma}\,\mathrm{TRANS}_4 \tag{8.32}$$

which mimics $\mathcal{D}$. We call transformation steps that are of this or of a similar form *$\{\mathrm{TRANS}_k\}_k$-amalgamation steps.*

The possibility to carry out such amalgamation steps makes it possible to simplify derivations in $\mathbf{HB}^{=}_{\mathbf{0}} + \mathrm{SYMM} + \{\mathrm{TRANS}_k\}_k$ in the way as stated by the following proposition.

Proposition 8.2.10 (Amalgamating $\{\mathbf{TRANS}_k\}_k$-applications). *Every derivation $\mathcal{D}$ in $\mathbf{HB}^{=}_{\mathbf{0}} + {}^{*}(\mu{-}\bot)^{\bot\mathrm{der}}_{l/r} + \mathrm{SYMM} + \{\mathrm{TRANS}_k\}_k$ can effectively be transformed, by applying $\{\mathrm{TRANS}_k\}_k$-amalgamation steps, into a derivation $\tilde{\mathcal{D}}$ in $\mathbf{HB}^{=}_{\mathbf{0}} +$ $+ {}^{*}(\mu - \bot)^{\bot der}_{l/r} + \mathrm{SYMM} + \{\mathrm{TRANS}_k\}_k$ with the same conclusion, with the same open assumption classes as $\mathcal{D}$, and with the property that no immediate subderivation leading up to a premise of an applications of a rule TRANS_{k_1} in $\tilde{\mathcal{D}}$, for some $k_1 \in \omega\backslash\{0\}$, has a bottommost application of a rule TRANS_{k_2}, for any $k_2 \in \omega\backslash\{0\}$.*

Sketch of the Proof. The assertion of the lemma can be shown by induction on the size $\mathrm{s}(\mathcal{D})$ of a derivation $\mathcal{D}$ in $\mathbf{HB}^{=}_{\mathbf{0}} + {}^{*}(\mu{-}\bot)^{\bot\mathrm{der}}_{l/r} + \mathrm{SYMM} + \{\mathrm{TRANS}_k\}_k$.

The induction step is straightforward and only one situation requires some attention. Namely, the case that $\mathcal{D}$ ends with an application of TRANS_{k_1}, for some $k_1 \in \omega \backslash \{0\}$, and that additionally one immediate subderivation of $\mathcal{D}$ also ends with a rule TRANS_{k_2}, for some $k_2 \in \omega \backslash \{0\}$ (for example, the derivation (8.31) is of such form). Here a $\{\mathrm{TRANS}_k\}_k$-amalgamation step at the bottom of $\mathcal{D}$ has to be carried out first with a resulting derivation $\mathcal{D}^{(1)}$ that ends with an application of $\mathrm{TRANS}_{k_1+k_2}$, that mimics $\mathcal{D}$, and that is of smaller size than $\mathcal{D}$ (for an example, we refer again to the step between the derivations in (8.31) and (8.32)). Then the induction hypothesis can be applied to $\mathcal{D}^{(1)}$ and in this way a desired derivation $\tilde{\mathcal{D}}$ is reached. □

In the tentative outline given above of the transformation developed here, we have used the somewhat imprecise expression "$\mathbf{HB}_{\mathbf{0}}^{=}$-end-part", which was designated by $\lceil \mathcal{D} \rceil_{\mathbf{HB}_{\mathbf{0}}^{=}}$, of a derivation in $\mathbf{HB}_{\mathbf{0}}^{=} + \mathrm{SYMM} + \mathrm{TRANS}$. Now we define two precise notions derived from this concept: the $\mathbf{HB}_{\mathbf{0}}^{=}$-end-derivation-context, and the $\mathbf{HB}_{\mathbf{0}}^{=}$-end-derivation of a derivation $\mathcal{D}$ in the extension of $\mathbf{HB}_{\mathbf{0}}^{=}$ with the additional rules introduced above. For given derivations $\mathcal{D}$ of this kind, the $\mathbf{HB}_{\mathbf{0}}^{=}$-end-derivation-context, and the $\mathbf{HB}_{\mathbf{0}}^{=}$-end-derivation are the respective results of first cutting off all subderivations of $\mathcal{D}$ that lead up to conclusions of applications of non-$\mathbf{HB}_{\mathbf{0}}^{=}$-rules (but keeping the respective conclusions), and of then replacing all formulas at the top originating from such conclusions by distinct context-holes, and respectively, by distinct marked assumptions.

Definition 8.2.11. ($\mathbf{HB}_{\mathbf{0}}^{=}$-end-derivations, $\mathbf{HB}_{\mathbf{0}}^{=}$-end-derivation-contexts of $\mathbf{HB}_{\mathbf{0}}^{=} + {}^{*}(\mu - \bot)_{l/r}^{\bot\mathbf{der}} + \mathbf{SYMM} + \mathbf{TRANS} + \{\mathbf{TRANS}_k\}_k$-derivations). Let $\mathcal{D}$ be a derivation in $\mathbf{HB}_{\mathbf{0}}^{=} + {}^{*}(\mu - \bot)_{l/r}^{\bot\mathrm{der}} + \mathrm{SYMM} + \mathrm{TRANS} + \{\mathrm{TRANS}_k\}_k$ that can be represented as of the form

$$\begin{array}{c}
\mathrm{TRANS}_{k_i}\ \dfrac{\begin{array}{cccc}\mathcal{D}_{i1} & \mathcal{D}_{i2} & & \mathcal{D}_{i(k_i+1)}\\ \tau' = \rho_1 & \rho'_1 = \rho_2 & \ldots & \rho'_{k_i} = \sigma'\end{array}}{\{\ (\tau_i = \sigma_i)_i\ \}_{i \in I_1}} \qquad \dfrac{\begin{array}{c}\mathcal{D}_{i1}\\ \sigma_i = \tau_i\end{array}}{\{\ (\tau_i = \sigma_i)_i\ \}_{i \in I_2}}\ \mathrm{SYMM}\\
\mathcal{DC}\\
\tau = \sigma
\end{array} \tag{8.33}$$

where the following holds:

(i) $I_1, I_2 \subseteq \omega$, $I_1 \cap I_2 = \emptyset$, and $I_1 \cup I_2 = \{1, \ldots, m\}$ for some $m \in \omega$ (if $m = 0$ then $I_1 = I_2 = \emptyset$),

(ii) for all $i \in I_1$, $k_i \in \omega \backslash \{0\}$, $\tau_i, \sigma_i, \rho_{i1}, \rho'_{i1}, \ldots, \rho_{i(k_i+1)}, \rho'_{i(k_i+1)} \in \mu Tp$ and $\mathcal{D}_{i1}$, $\ldots$, $\mathcal{D}_{i(k_i+1)}$ are derivations in $\mathbf{HB}_{\mathbf{0}}^{=} + {}^{*}(\mu - \bot)_{l/r}^{\bot\mathrm{der}} + \mathrm{TRANS} + \mathrm{SYMM} + {} + \{\mathrm{TRANS}_k\}_k$, and for all $i \in I_2$, $\tau_i, \sigma_i \in \mu Tp$ and $\mathcal{D}_{i1}$ is a derivation in $\mathbf{HB}_{\mathbf{0}}^{=} + {}^{*}(\mu - \bot)_{l/r}^{\bot\mathrm{der}} + \mathrm{SYMM} + \mathrm{TRANS} + \{\mathrm{TRANS}_k\}_k$, and

(iii) $\mathcal{DC} \in \mathcal{D}er\mathcal{C}txt_m(\mathbf{HB}^{=}_{\mathbf{0}})$, in which every context-hole $[\,]_i$, for $i \in \{1,\ldots,m\}$ occurs precisely once, and the context-holes $[\,]_1$, ..., $[\,]_m$ occur at the top of $\mathcal{DC}$ ordered in a traversal from left to right in the labeled tree underlying $\mathcal{DC}$.

Then we say that the derivation-context $\mathcal{DC}$ is the $\mathbf{HB}^{=}_{\mathbf{0}}$*-end-derivation-context of* $\mathcal{D}$, and we designate $\mathcal{DC}$ by $\lceil\mathcal{D}\rceil^{\{[\,]_i\}_i}_{\mathbf{HB}^{=}_{\mathbf{0}}}$.

Let furthermore $\{u_i\}_{i=1,\ldots,m}$ be a family of distinct assumption markers that do not occur in $\mathcal{D}$. Then we call the derivation

$$\begin{array}{c} \{\ (\,(\tau_i = \sigma_i)^{u_i}\,)_i\ \}_{i \in I_1 \cup I_2} \\ \mathcal{DC} \\ \tau = \sigma \end{array} \tag{8.34}$$

in $\mathbf{HB}^{=}_{\mathbf{0}}$, which arises as the result $\mathcal{DC}[(\tau_1 = \sigma_1)^{u_1}]_1 \ldots [(\tau_m = \sigma_m)^{u_m}]_m$ of filling the marked assumptions $(\tau_1 = \sigma_1)^{u_1}$, ..., $(\tau_m = \sigma_m)^{u_m}$ into the context-holes $[\,]_1$, ..., $[\,]_m$ of $\mathcal{DC}$ respectively, a $\mathbf{HB}^{=}_{\mathbf{0}}$*-end-derivation of* $\mathcal{D}$, and we designate it $\lceil\mathcal{D}\rceil^{\{u_i\}_i}_{\mathbf{HB}^{=}_{\mathbf{0}}}$.

Similarly, if $\mathcal{D}$ is a derivation in $\mathbf{HB}^{=}_{\mathbf{0}} + {}^{*}(\mu - \bot)^{\bot\text{der}}_{l/r} + \text{SYMM} + \text{TRANS} + {}+ \{\text{TRANS}_k\}_k$ of the form (8.33) for which the conditions (i), (ii) and (instead of (iii))

(iii)′ $\mathcal{DC} \in \mathcal{D}er\mathcal{C}txt_m(\mathbf{HB}^{=}_{\mathbf{0}})$, in which every context-hole $[\,]_i$, for $i \in \{1,\ldots,m\}$ occurs precisely once, and the context-holes $[\,]_1$, ..., $[\,]_m$ occur at the top of $\mathcal{DC}$ ordered in a traversal from left to right in the labeled tree underlying $\mathcal{DC}$.

hold, then we say that the derivation-context $\mathcal{DC}$ is the *(*$\mathbf{HB}^{=}_{\mathbf{0}} + {}^{*}(\mu - \bot)^{\bot der}_{l/r}$*)-end-derivation-context of* $\mathcal{D}$.

⊠

As a further auxiliary notion that is needed for the formulation of a number of lemmas below, we define the "$\mathbf{HB}^{=}_{\mathbf{0}}$-height" of derivations that contain also rules which are not part of the system $\mathbf{HB}^{=}_{\mathbf{0}}$.

Definition 8.2.12 ($\mathbf{HB}^{=}_{\mathbf{0}}$-height of derivations containing non-$\mathbf{HB}^{=}_{\mathbf{0}}$-rules). Let $\mathcal{D}$ be a derivation in $\mathbf{HB}^{=}_{\mathbf{0}} + \text{SYMM} + \text{TRANS} + \{\text{TRANS}_k\}_k + {}^{*}(\mu - \bot)^{\bot\text{der}}_{l/r}$ that contains at least one application of a rule that is not contained in $\mathbf{HB}^{=}_{\mathbf{0}}$.

By the $\mathbf{HB}^{=}_{\mathbf{0}}$*-height* $h_{\mathbf{HB}^{=}_{\mathbf{0}}}(\mathcal{D})$ of $\mathcal{D}$ we mean the minimal height in $\mathcal{D}$ of the conclusion of an application of SYMM, TRANS, ${}^{*}(\mu - \bot)^{\bot\text{der}}_{l/r}$, or of a rule from $\{\text{TRANS}_k\}_k$ in $\mathcal{D}$, i.e. the minimal number of rule applications that are passed in $\mathcal{D}$ from the conclusion of an application of a non-$\mathbf{HB}^{=}_{\mathbf{0}}$-rule on a thread downwards to the conclusion of $\mathcal{D}$. ⊠

We are now able to formulate as the following lemma the main auxiliary assertion for step two in our transformation (cf. (St2) in the tentative overview given earlier); it will eventually be used to justify step (St2)′ in our proof of Theorem 8.2.2 on page 318.

Lemma 8.2.13. *Let $\tau, \sigma \in \mu Tp$, and let $\mathcal{D}$ be a derivation in $\mathbf{HB}_{\mathbf{0}}^{=} + \mathrm{SYMM} + {} + \{\mathrm{TRANS}_k\}_k$ with conclusion $\tau = \sigma$ and without open assumptions.*

Then a sequence $\mathcal{SD} = \langle \mathcal{D}^{(n)} \rangle_{n \in I}$ of derivations in the system $\mathbf{HB}_{\mathbf{0}}^{=} + \mathrm{SYMM} + {} + \{\mathrm{TRANS}_k\}_k$ can effectively be built, where the index set I is either ω or of the form $[0, n_{\max}] \cap \omega$, for some $n_{\max} \in \omega$ (hence $0 \in I$ and $\mathcal{SD}$ is either finite or countably infinite, but not empty), with the following properties:

- *(i) the derivation $\mathcal{D}^{(0)}$ is the derivation $\mathcal{D}$ (i.e. the sequence $\mathcal{SD}$ starts with $\mathcal{D}^{(0)}$);*
- *(ii) for all $n \in \omega$, the derivation $\mathcal{D}^{(n)}$ has conclusion $\tau = \sigma$ and does not contain open assumptions (and hence as a consequence of this and of (i), all $\mathcal{D}^{(n)}$ mimic $\mathcal{D}$);*
- *(iii) for all $n \in I \setminus \{0\}$, $\mathcal{D}^{(n)}$ does not contain successive applications of* REN, *nor nlμb-decreasing applications of* $\mathrm{FOLD}_{l/r}$ *(see Definition C.9, Appendix C);*
- *(iv) for all $n \in I$ such that $n + 1 \in I$, the derivation $\mathcal{D}^{(n+1)}$ is the result of applying an effective proof-theoretic transformation to the derivation $\mathcal{D}^{(n)}$;*
- *(v) if $\mathcal{SD}$ is finite, i.e. if $I = \{0, 1, \ldots, n_{\max}\}$ and $\mathcal{SD} = \langle \mathcal{D}^{(1)}, \ldots, \mathcal{D}^{(n_{\max})} \rangle$ with some $n_{\max} \in \omega$, then all derivations $\mathcal{D}^{(n)}$ in $\mathcal{SD}$ with $n \in \{0, \ldots, n_{\max} - 1\}$ contain applications of* SYMM *and/or of rules from* $\{\mathrm{TRANS}_k\}_k$, *in contrast with the last derivation $\mathcal{D}^{(n_{\max})}$ of $\mathcal{SD}$, which is a derivation in $\mathbf{HB}_{\mathbf{0}}^{=}$;*
- *(vi) if $\mathcal{SD}$ is infinite, i.e. if $\mathcal{SD} = \langle \mathcal{D}^{(n)} \rangle_{n \in \omega}$ holds, then all derivations in $\mathcal{SD}$ contain applications of* SYMM *and/or* TRANS, *and furthermore,*

$$\langle h_{\mathbf{HB}_{\mathbf{0}}^{=}}(\mathcal{D}^{(n)}) \rangle_n \longrightarrow_{n \to \infty} \infty \tag{8.35}$$

holds, i.e. the $\mathbf{HB}_{\mathbf{0}}^{=}$-heights of $\mathcal{D}^{(n)}$, and thus the minimal heights of SYMM- *or* $+\{\mathrm{TRANS}_k\}_k$*-applications in $\mathcal{D}^{(n)}$, tend to infinity.*

The *proof* of this lemma will be given below on page 314. In this proof a number of further lemmas will be used that formulate assertions about particular forms of derivations in $\mathbf{HB}_{\mathbf{0}}^{=} + {}^{*}(\mu{-}\bot)^{\bot \mathrm{der}}_{l/r} + \mathrm{SYMM} + \{\mathrm{TRANS}_k\}_k$, which forms can be produced by using primarily the operations of upwards-permuting of applications of SYMM as well as of rules in $\{\mathrm{TRANS}_k\}_k$, and of 'unfolding' derivations appropriately.

As an aside, we want to mention that our proof of Lemma 8.2.13 will actually show a slightly more specific statement concerning the proof-theoretic transformation between successive derivations $\mathcal{D}^{(n)}$ and $\mathcal{D}^{(n+1)}$ belonging to the sequence $\mathcal{SD} = \langle \mathcal{D}^{(n)} \rangle_{n \in I}$ in the statement of this lemma: our proof is in fact able to demonstrate the statement which results from the assertion of the lemma by replacing property (iv) there by the property

(iv)* for all $n \in I$ such that $n + 1 \in I$, the derivation $\mathcal{D}^{(n+1)}$ is the result of applying to the derivation $\mathcal{D}^{(n)}$ an effective proof-theoretic transformation that uses only the steps (a)–(i) mentioned in Lemma 8.2.19 below (and itemized there as well as in Lemma 8.2.15, Lemma 8.2.16, and in Lemma 8.2.18).

We state and prove the lemmas required for the proof of Lemma 8.2.13 successively below. We start with a rather easy statement about a special form of derivations in $\mathbf{HB}^{=}_{\mathbf{0}} + {}^{*}(\mu{-}\bot)^{\bot\text{der}}_{l/r} + \text{SYMM} + \{\text{TRANS}_k\}_k$ that can be reached by upwards-permutation movements of applications of SYMM.

Lemma 8.2.14. *Every derivation $\mathcal{D}$ in the system $\mathbf{HB}^{=}_{\mathbf{0}} + {}^{*}(\mu{-}\bot)^{\bot\text{der}}_{l/r} + \text{SYMM} + {}$ ${}+ \{\text{TRANS}_k\}_k$ can be transformed into a derivation $\tilde{\mathcal{D}}$ in $\mathbf{HB}^{=}_{\mathbf{0}} + {}^{*}(\mu-\bot)^{\bot\text{der}}_{l/r} + {}$ ${}+ \text{SYMM} + \{\text{TRANS}_k\}_k$ with the same conclusion, with the same open assumption classes as $\mathcal{D}$, and such that the derivation $\tilde{\mathcal{D}}$ contains applications of* SYMM *only as single applications immediately below marked assumptions.*

Sketch of the Proof. The following three facts are either entirely obvious (those in items (a) and (b)) or easy to verify (the fact in item (c)):

(a) Applications of SYMM that immediately follow axioms of $\mathbf{HB}^{=}_{\mathbf{0}}$ (i.e. that follow axioms (REFL)) can be removed.

(b) Any two successive applications of SYMM can be removed.

(c) Applications of SYMM can be permuted upwards over every application of a rule in $\mathbf{HB}^{=}_{\mathbf{0}} + \{\text{TRANS}_k\}_k$.

These three observations are sufficient to show the assertion of the lemma by a straightforward induction on the depth $|\mathcal{D}|$ of derivations $\mathcal{D}$ in $\mathbf{HB}^{=}_{\mathbf{0}} + \text{SYMM} + {}$ ${}+ \text{TRANS}$ with a subinduction on the number of applications of ARROW/FIX in $\mathcal{D}$. Hereby the subinduction is invoked only invoked in the induction step for the case that a bottommost application of SYMM has to be permuted upwards over a preceding application of ARROW/FIX. □

Next we formulate and prove a lemma about transforming an arbitrary derivation in $\mathbf{HB}^{=}_{\mathbf{0}} + {}^{*}(\mu{-}\bot)^{\bot\text{der}}_{l/r} + \text{SYMM} + \{\text{TRANS}_k\}_k$ that contains only a single application of TRANS_1, which occurs at the bottom, into a special form by permuting this application of TRANS_1 upwards as far as possible.

Lemma 8.2.15. *Let $\mathcal{D}$ be a derivation in $\mathbf{HB}^{=}_{\mathbf{0}} + {}^{*}(\mu{-}\bot)^{\bot\text{der}}_{l/r} + \text{SYMM} + \text{TRANS}_1$ that is of the form*

$$\frac{\begin{array}{cc}\mathcal{D}_1 & \mathcal{D}_2\\ \tau' = \rho & \rho' = \sigma'\end{array}}{\tau = \sigma}\,\text{TRANS}_1\,, \tag{8.36}$$

where $\mathcal{D}_1$ and $\mathcal{D}_2$ are derivations in $\mathbf{HB}^{=}_{\mathbf{0}} + {}^{*}(\mu{-}\bot)^{\bot\text{der}}_{l/r} + \text{SYMM}$ (and that therefore do not contain applications of TRANS_1*), and that fulfills the following condition:*

$$\left.\begin{array}{l}\textit{all open marked assumptions of } \mathcal{D} \textit{ are equations between composite}\\ \textit{recursive types, i.e. they are of the form } (\chi_{11} \to \chi_{12} = \chi_{21} \to \chi_{22})^{u}\\ \textit{for respective } \chi_{11}, \chi_{12}, \chi_{21}, \chi_{22} \in \mu Tp\textit{, and assumption markers } u.\end{array}\right\} \tag{8.37}$$

Then $\mathcal{D}$ can effectively be transformed, by

(a) *upwards-permutation steps of* SYMM *and* TRANS_1*-applications,*

(b) *easy replacement-steps concerned with the presence of axioms in premises of* SYMM- *and* TRANS_1*-applications,*

(c) *elimination-steps of pairs of successive* SYMM*-applications, and by*

into a derivation $\mathcal{D}'$ *in* $\mathbf{HB}^{=}_{\mathbf{0}} + {}^*(\mu-\bot)^{\bot\mathrm{der}}_{l/r} + \mathrm{SYMM} + \mathrm{TRANS}_1$ *of the form*

$$\begin{array}{c}
\mathrm{TRANS}_1\,\dfrac{\begin{array}{cc}\mathcal{D}_{i1} & \mathcal{D}_{i2}\\ \tau'_{i1}\to\tau'_{i2}=\rho_{i1}\to\rho_{i2} & \rho'_{i1}\to\rho'_{i2}=\sigma'_{i1}\to\sigma'_{i2}\end{array}}{\{\ (\,\tau_{i1}\to\tau_{i2}=\sigma_{i1}\to\sigma_{i2}\,)_i\ \}_{i\in I_1}} \qquad \dfrac{\begin{array}{c}(\mathrm{Assm})\\ (\sigma_{i1}\to\sigma_{i2}=\tau_{i1}\to\tau_{i2})^{u_i}\end{array}}{\{\ (\,\tau_{i1}\to\tau_{i2}=\sigma_{i1}\to\sigma_{i2}\,)_i\ \}_{i\in I_2}}\,\mathrm{SYMM}\\
\mathcal{DC}\\
\tau=\sigma
\end{array} \tag{8.38}$$

where

(i) $I_1, I_2 \subseteq \omega$*,* $I_1 \cap I_2 = \emptyset$*, and* $I_1 \cup I_2 = \{1,\ldots,m\}$ *for some* $m \in \omega$ *(if* $m = 0$ *then* $I_1 = I_2 = \emptyset$*),*

(ii) $\mathcal{DC} \in \mathcal{D}er\mathcal{C}txt_m(\mathbf{HB}^{=}_{\mathbf{0}} + {}^*(\mu-\bot)^{\bot\mathrm{der}}_{l/r})$ *(in particular,* $\mathcal{DC}$ *contains no applications of* SYMM *and* TRANS_1*) is the (*$\mathbf{HB}^{=}_{\mathbf{0}} + {}^*(\mu-\bot)^{\bot der}_{l/r}$*)-end-derivation-context of* $\mathcal{D}'$*,*

(iii) *all derivations* $\mathcal{D}_{ij}$*, for* $i \in I_1$ *and* $j \in \{1,2\}$*, are derivations in* $\mathbf{HB}^{=}_{\mathbf{0}} + {} + \mathrm{SYMM} + \mathrm{TRANS}_1 + {}^*(\mu-\bot)^{\bot\mathrm{der}}_{l/r}$ *that are of one of the four possible forms*

$$\begin{array}{c}(\mathrm{Assm})\\ (\tilde\tau=\tilde\sigma)^{\tilde u}\end{array} \qquad \dfrac{\begin{array}{c}(\mathrm{Assm})\\ (\tilde\sigma=\tilde\tau)^{\tilde u}\end{array}}{\tilde\tau=\tilde\sigma}\,\mathrm{SYMM} \qquad \begin{array}{c}\mathcal{D}_{ij}\\ \cdots\cdots\cdots\\ \tilde\tau=\tilde\sigma\end{array}\ \mathrm{ARROW} \qquad \begin{array}{c}[\tilde\tau=\tilde\sigma]^{\tilde u}\\ \mathcal{D}_{ij}\\ \cdots\cdots\cdots\\ \tilde\tau=\tilde\sigma\end{array}\ \mathrm{ARROW/FIX},\ \tilde u \tag{8.39}$$

for respective $\tilde\tau_1, \tilde\tau_2, \tilde\sigma_1, \tilde\sigma_2 \in \mu Tp$ *and assumption markers* $\tilde u$ *(i.e. each* $\mathcal{D}_{ij}$ *is a derivation consisting of a marked assumption, or of a marked assumption that is immediately followed by an application of* SYMM*, or it is a derivations that ends with an application of* ARROW *or* ARROW/FIX*), and where, for all* $i \in I_1$*, not both of* $\mathcal{D}_{i1}$ *and* $\mathcal{D}_{i2}$ *end with an application of* ARROW*,*

such that $\mathcal{D}'$ *has the same conclusion and the same open assumption classes as* $\mathcal{D}$*.*

Proof. In view of Lemma 8.2.14 if suffices to prove the lemma for all such derivations in $\mathbf{HB}^{=}_{\mathbf{0}} + {}^*(\mu-\bot)^{\bot\mathrm{der}}_{l/r} + \mathrm{SYMM} + \mathrm{TRANS}_1$ of the form (8.36) with (8.37) fulfilled, where $\mathcal{D}_1$ and $\mathcal{D}_2$ are derivations in $\mathbf{HB}^{=}_{\mathbf{0}} + {}^*(\mu-\bot)^{\bot\mathrm{der}}_{l/r} + \mathrm{SYMM}$ that contain applications of SYMM only immediately below marked assumptions as single applications. This is because for every derivation $\mathcal{D}$ in $\mathbf{HB}^{=}_{\mathbf{0}} + {}^*(\mu-\bot)^{\bot\mathrm{der}}_{l/r} + {} + \mathrm{SYMM} + \mathrm{TRANS}_1$ of the form (8.36), where $\mathcal{D}_1$ and $\mathcal{D}_2$ are derivations in $\mathbf{HB}^{=}_{\mathbf{0}} + {} + {}^*(\mu-\bot)^{\bot\mathrm{der}}_{l/r}$, $\mathcal{D}_1$ and $\mathcal{D}_2$ can first be transformed due to Lemma 8.2.14, by upwards-permutations of SYMM-applications and by removing successive applications of SYMM, into derivations $\tilde{\mathcal{D}}_1$ and $\tilde{\mathcal{D}}_2$ with the same conclusion and the same open assumption classes as, respectively, $\mathcal{D}_1$ and $\mathcal{D}_2$, and where $\tilde{\mathcal{D}}_1$ and $\tilde{\mathcal{D}}_2$ contain SYMM-applications only immediately below marked assumptions. In proving the described restricted statement, we will proceed by induction on $|\mathcal{D}_1| + |\mathcal{D}_2|$.

For this, we let therefore $\mathcal{D}$ be a derivation in $\mathbf{HB}^{=}_{\mathbf{0}} + {}^{*}(\mu{-}\bot)^{\bot\text{der}}_{l/r} + \text{SYMM} + {} + \text{TRANS}_1$ of the form (8.36) with (8.37) fulfilled, where $\mathcal{D}_1$ and $\mathcal{D}_2$ are such derivations in $\mathbf{HB}^{=}_{\mathbf{0}} + \text{SYMM} + {}^{*}(\mu{-}\bot)^{\bot\text{der}}_{l/r}$ that contain SYMM-applications only immediately below marked assumptions as single applications. We have to show that $\mathcal{D}$ can be transformed, by steps of the kind (a)–(c) as described in the lemma, into a derivation $\mathcal{D}'$ of the form (8.38) that fulfills the conditions (i)–(iii) in the lemma and that has the same conclusion and the same open assumption classes as $\mathcal{D}$. We will do so by case-distinction on the form of the subderivations $\mathcal{D}_1$ and $\mathcal{D}_2$ of $\mathcal{D}$, and in particular by distinguishing cases according to the last rule applications in $\mathcal{D}_1$ and $\mathcal{D}_2$, if such are present.

If one of $\mathcal{D}_1$ and $\mathcal{D}_2$ is an axiom (REFL), then for $\mathcal{D}'$ the result of appending an application of REN to the other derivation (out of $\mathcal{D}_1$ and $\mathcal{D}_2$) can be chosen.

If both of $\mathcal{D}_1$ and $\mathcal{D}_2$ consist of marked assumptions or a marked assumption that is respectively followed by an application of SYMM, then $\mathcal{D}$ is itself already of the required form.

Next, let us now consider the case in which one of $\mathcal{D}_1$ or $\mathcal{D}_2$ consists either of a marked assumption or ends with an application of SYMM while the other derivation is not an axiom nor a marked assumption, but ends with an application of a $\mathbf{HB}^{=}_{\mathbf{0}} + {}^{*}(\mu{-}\bot)^{\bot\text{der}}_{l/r}$-rule. For example, we consider the case that $\mathcal{D}$ is of the form

$$\dfrac{\text{SYMM}\;\dfrac{(\rho_1 \to \rho_2 = \tau_1' \to \tau_2')^u}{\tau_1' \to \tau_2' = \rho_1 \to \rho_2} \qquad \dfrac{\begin{matrix}\mathcal{D}_{21}\\ \tilde{\rho} = \tilde{\sigma}\end{matrix}}{\rho_1' \to \rho_2' = \sigma'}\;R}{\tau_1 \to \tau_2 = \sigma}\;\text{TRANS}_1 \tag{8.40}$$

with some $\tau_1, \tau_2, \tau_1', \tau_2', \rho_1, \rho_2, \rho_1', \rho_2', \tilde{\rho}, \sigma, \tilde{\sigma} \in \mu Tp$ and an assumption marker u, where R means the name of a rule in $\mathbf{HB}^{=}_{\mathbf{0}} + {}^{*}(\mu{-}\bot)^{\bot\text{der}}_{l/r}$ (since we have assumed (8.37) on $\mathcal{D}$, the marked assumption $(\rho_1 \to \rho_2 = \tau_1' \to \tau_2')^u$ in $\mathcal{D}_1$ is not of a special form!); our treatment of this subcase can be carried over to all other subcases of the considered situation here. Due to the form of the conclusion of the displayed application of R (as an equation between composite recursive types), this can only be an application of ARROW, ARROW/FIX, FOLD_r, $(\mu{-}\bot)^{\bot\text{der}}_{r}$, ${}^{*}(\mu{-}\bot)^{\bot\text{der}}_{r}$, or REN. If it is an application of ARROW or ARROW/FIX, then $\mathcal{D}$ is already itself of the desired form. If the displayed application of R is one of FOLD_r, $(\mu{-}\bot)^{\bot\text{der}}_{r}$, ${}^{*}(\mu{-}\bot)^{\bot\text{der}}_{r}$, or REN, then $\mathcal{D}_{21}$ has conclusion $\rho_1'' \to \rho_2'' = \tilde{\sigma}$ for some $\rho_1'', \rho_2'' \in \mu Tp$ such that $\rho_1'' \to \rho_2''$ is a variant[6] of $\rho_1 \to \rho_2$ and of $\rho_1' \to \rho_2'$. Therefore it can be transformed, by permuting the application of TRANS_1 upwards over the application

[6] If the displayed application of R is an application of FOLD_r, $(\mu{-}\bot)^{\bot\text{der}}_{r}$, or ${}^{*}(\mu{-}\bot)^{\bot\text{der}}_{r}$, then ρ_1'' and ρ_2'' are actually equal to ρ_1' and ρ_2', respectively.

of R at the bottom of $\mathcal{D}_2$, into the derivation $\tilde{\mathcal{D}}$ of the form

$$\dfrac{\dfrac{\text{SYMM}\,\dfrac{(\rho_1\rightarrow\rho_2=\tau_1'\rightarrow\tau_2')^u}{\tau_1'\rightarrow\tau_2'=\rho_1\rightarrow\rho_2} \qquad \begin{array}{c}\mathcal{D}_{21}\\ \rho_1''\rightarrow\rho_2''=\tilde{\sigma}\end{array}}{\tau_1\rightarrow\tau_2=\tilde{\sigma}}\,\text{TRANS}_1}{\tau_1\rightarrow\tau_2=\sigma}\,R \tag{8.41}$$

in which the induction hypothesis can be applied to the subderivation $\tilde{\mathcal{D}}_1$ ending with the TRANS_1-application (since $1+|\mathcal{D}_{21}|<1+|\mathcal{D}_2|=|\mathcal{D}_1|+|\mathcal{D}_2|$). Therefore $\tilde{\mathcal{D}}_1$ can be transformed into a derivation $\tilde{\mathcal{D}}_1'$ of the required form. And by replacing $\tilde{\mathcal{D}}_1$ in $\tilde{\mathcal{D}}$ by $\tilde{\mathcal{D}}_1'$, we eventually arrive at a derivation $\mathcal{D}'$ in $\mathbf{HB}^{=}_{\overline{\mathbf{0}}}+{}^{*}(\mu-\bot)^{\bot\text{der}}_{l/r}+$ $+\,\text{SYMM}+\text{TRANS}_1$ of the required form that furthermore has the same conclusion and the same open assumption classes as $\mathcal{D}$.

If one of $\mathcal{D}_1$ or $\mathcal{D}_2$ ends with an application of ARROW/FIX, while the other terminates with an application of a $\mathbf{HB}^{=}_{\overline{\mathbf{0}}}+{}^{*}(\mu{-}\bot)^{\bot\text{der}}_{l/r}$-rule other than ARROW/FIX and ARROW, then it can be argued analogously as in the previous case above. If one of $\mathcal{D}_1$ and $\mathcal{D}_2$ end with an application of ARROW/FIX, and the other is a marked assumption or it ends with an application of SYMM, of ARROW, or of ARROW/FIX, then $\mathcal{D}$ is already itself of the required form.

If, however, both $\mathcal{D}_1$ and $\mathcal{D}_2$ are derivations that end with an application of a $\mathbf{HB}^{=}_{\overline{\mathbf{0}}}+{}^{*}(\mu{-}\bot)^{\bot\text{der}}_{l/r}$-rule other than ARROW/FIX, then the application of TRANS_1 at the bottom of $\mathcal{D}$ can be permuted upwards over one or over both of the immediately preceding rule applications as stated and described by Lemma 8.2.7 with the result of a derivation $\tilde{\mathcal{D}}$. Then the induction hypothesis can be applied to the subderivation(s) of $\tilde{\mathcal{D}}$ ending with the newly arisen TRANS_k-application(s) (according to Lemma 8.2.7 there are at most two such subderivations), thereby giving derivations of the required form, which can be combined (according to the position of the mentioned subderivations in $\tilde{\mathcal{D}}$) to reach a derivation $\mathcal{D}'$ in $\mathbf{HB}^{=}_{\overline{\mathbf{0}}}+{}^{*}(\mu-\bot)^{\bot\text{der}}_{l/r}+$ $+\,\text{SYMM}+\{\text{TRANS}_k\}_k$ of the required form and with the same conclusion and the same open assumption classes as $\mathcal{D}$.

In this way we have considered all possible cases for the subderivations $\mathcal{D}_1$ and $\mathcal{D}_2$ and have carried out the induction step for the statement we needed to prove.

□

The next lemma is a generalization of Lemma 8.2.15: it states that every derivation $\mathcal{D}$ in $\mathbf{HB}^{=}_{\overline{\mathbf{0}}}+{}^{*}(\mu-\bot)^{\bot\text{der}}_{l/r}+\text{SYMM}+\{\text{TRANS}_k\}_k$ with a bottommost application of TRANS_1, which is not necessarily the single application of TRANS_1 in $\mathcal{D}$, can be transformed into a similar special form as stated by Lemma 8.2.15. The transformation applied hereby proceeds primarily by permuting applications of TRANS_1 upwards, by ($\{\text{TRANS}_k\}_k \rightsquigarrow \text{TRANS}_1$)-mimicking steps, and by $\{\text{TRANS}_k\}_k$-amalgamation steps.

Lemma 8.2.16. *Let $\mathcal{D}$ be a derivation in the extension* $\mathbf{HB}^{=}_{\overline{\mathbf{0}}}+{}^{*}(\mu-\bot)^{\bot\text{der}}_{l/r}+$

$+\,\mathrm{SYMM}+\{\mathrm{TRANS}_k\}_k$ *of* $\mathbf{HB}_{\mathbf{0}}^{=}$ *such that* $\mathcal{D}$ *is of the form*

$$\frac{\begin{array}{cc}\mathcal{D}_1 & \mathcal{D}_2\\ \tau' = \rho & \rho' = \sigma'\end{array}}{\tau = \sigma}\;\mathrm{TRANS}_1 \tag{8.42}$$

where $\mathcal{D}_1$ *and* $\mathcal{D}_2$ *are derivations in* $\mathbf{HB}_{\mathbf{0}}^{=} + {}^{*}(\mu\text{–}\bot)^{\bot\mathrm{der}}_{l/r} + \mathrm{SYMM} + \{\mathrm{TRANS}_k\}_k$, *too, and such that* $\mathcal{D}$ *fulfills the condition* (8.37) *on its open marked assumptions.*

Then $\mathcal{D}$ *can effectively be transformed, by steps (a)–(c) as described above in Lemma 8.2.15 and additionally by*

(d) *(*$\{\mathrm{TRANS}_k\}_k \rightsquigarrow \mathrm{TRANS}_1$*)-mimicking steps,*

(e) $\{\mathrm{TRANS}_k\}_k$*-amalgamation steps,*

into a derivation $\mathcal{D}'$ *in* $\mathbf{HB}_{\mathbf{0}}^{=} + {}^{*}(\mu\text{–}\bot)^{\bot\mathrm{der}}_{l/r} + \mathrm{SYMM} + \{\mathrm{TRANS}_k\}_k$ *of the form*

$$\begin{array}{c}
\mathrm{TRANS}_{k_i}\,\dfrac{\mathcal{D}_{i1}\quad \mathcal{D}_{i2}\quad \ldots\quad \mathcal{D}_{i(k_i+1)}}{\{\ (\,\tau_{i1} \to \tau_{i2} = \sigma_{i1} \to \sigma_{i2}\,)_i\ \}_{i\in I_1}}
\qquad
\dfrac{\begin{array}{c}(\mathrm{Assm})\\ (\sigma_{i1} \to \sigma_{i2} = \tau_{i1} \to \tau_{i2})^{u_i}\end{array}}{\{\ (\,\tau_{i1} \to \tau_{i2} = \sigma_{i1} \to \sigma_{i2}\,)_i\ \}_{i\in I_2}}\,\mathrm{SYMM}\\
\mathcal{DC}\\
\tau = \sigma
\end{array} \tag{8.43}$$

where

(i) $I_1, I_2 \subseteq \omega$, $I_1 \cap I_2 = \emptyset$, *and* $I_1 \cup I_2 = \{1,\ldots,m\}$ *for some* $m \in \omega$ *(if* $m = 0$ *then* $I_1 = I_2 = \emptyset$*),*

(ii) *for all* $i \in I_1$, $k_i \in \omega\backslash\{0\}$, $\tau_{i1}, \tau_{i2}, \sigma_{i1}, \sigma_{i2} \in \mu Tp$, *and* $\mathcal{D}_{i1}$, $\mathcal{D}_{i2}$, $\ldots$, $\mathcal{D}_{i(k_i+1)}$ *are derivations in* $\mathbf{HB}_{\mathbf{0}}^{=} + {}^{*}(\mu\text{–}\bot)^{\bot\mathrm{der}}_{l/r} + \mathrm{SYMM} + \{\mathrm{TRANS}_k\}_k$, *and furthermore, for all* $i \in I_2$, u_i *is an assumption marker and* $\tau_{i1}, \tau_{i2}, \sigma_{i1}, \sigma_{i2} \in \mu Tp$,

(iii) $\mathcal{DC} \in \mathcal{D}er\mathcal{C}txt_m(\mathbf{HB}_{\mathbf{0}}^{=} + {}^{*}(\mu\text{–}\bot)^{\bot\mathrm{der}}_{l/r})$ *is the* $\mathbf{HB}_{\mathbf{0}}^{=} + {}^{*}(\mu\text{–}\bot)^{\bot\mathrm{der}}_{l/r}$*-end-derivation-context of* $\mathcal{D}$*,*

(iv) *all derivations* $\mathcal{D}_{i1}$, $\mathcal{D}_{i2}$, $\ldots$, $\mathcal{D}_{i(k_i+1)}$, *for* $i \in I_1$, *are such derivations in* $\mathbf{HB}_{\mathbf{0}}^{=} + {}^{*}(\mu\text{–}\bot)^{\bot\mathrm{der}}_{l/r} + \mathrm{SYMM} + \{\mathrm{TRANS}_k\}_k$ *that are of one of the four possible forms in* (8.39) *for respective* $\tilde{\tau}_1, \tilde{\tau}_2, \tilde{\sigma}_1, \tilde{\sigma}_2 \in \mu Tp$ *and assumption markers* $\tilde{u}$, *and where not all of* $\mathcal{D}_{i1}$, $\mathcal{D}_{i2}$, $\ldots$, $\mathcal{D}_{i(k_i+1)}$ *end with an application of* ARROW*,*

such that $\mathcal{D}'$ *has the same conclusion and the same open assumption classes as* $\mathcal{D}$*.*

Sketch of the Proof. In view of Lemma 8.2.15 it suffices to prove the following *restricted statement* of the lemma:

> For all derivations in $\mathbf{HB}^{=}_{\mathbf{0}} + {}^*(\mu\!-\!\perp)^{\perp\text{der}}_{l/r} + \text{SYMM} + \{\text{TRANS}_k\}_k$ of the form (8.42), where $\mathcal{D}_1$ and $\mathcal{D}_2$ are derivations in $\mathbf{HB}^{=}_{\mathbf{0}} + {}^*(\mu\!-\!\perp)^{\perp\text{der}}_{l/r} + \text{SYMM} + \{\text{TRANS}_k\}_k$ that are themselves of the form as described in the consequence of the lemma (i.e. that have the property that bottommost SYMM-applications which are not followed by $\{\text{TRANS}_k\}_k$-applications have marked assumptions in their premise, and that the immediate subderivations of bottommost $\{\text{TRANS}_k\}_k$-applications are of one of the four possible forms in (8.39), but do not all end with ARROW-applications) it holds that $\mathcal{D}$ can effectively be transformed, by steps (a)–(e) as mentioned in the lemma, into a derivation $\mathcal{D}'$ in $\mathbf{HB}^{=}_{\mathbf{0}} + {}^*(\mu\!-\!\perp)^{\perp\text{der}}_{l/r} + \text{SYMM} + \{\text{TRANS}_k\}_k$ of the form (8.43) with (i)–(iv) and with the same conclusion and the same open assumption classes as $\mathcal{D}$. (8.44)

This reduction of our proof obligation is due to the fact that the special case of the statement of the lemma with a derivation $\mathcal{D}$ in $\mathbf{HB}^{=}_{\mathbf{0}} + {}^*(\mu\!-\!\perp)^{\perp\text{der}}_{l/r} + \text{SYMM} + \{\text{TRANS}_k\}_k$ of the form (8.42), where $\mathcal{D}_1$ and $\mathcal{D}_2$ do not contain applications of $\{\text{TRANS}_k\}_k$, is settled by Lemma 8.2.15; and that, due to the restricted statement of the lemma, the (unrestricted) statement of the lemma follows, for every derivation $\mathcal{D}$ in $\mathbf{HB}^{=}_{\mathbf{0}} + {}^*(\mu\!-\!\perp)^{\perp\text{der}}_{l/r} + \text{SYMM} + \{\text{TRANS}_k\}_k$ of the form (8.42), by

- mimicking all TRANS_k-applications, for all respective $k \in \omega\backslash\{0\}$, in $\mathcal{D}_1$ and $\mathcal{D}_2$ by respectively k TRANS_1-applications with the result of $\tilde{\mathcal{D}}_1$ and $\tilde{\mathcal{D}}_2$ in $\mathbf{HB}^{=}_{\mathbf{0}} + {}^*(\mu\!-\!\perp)^{\perp\text{der}}_{l/r} + \text{SYMM} + \text{TRANS}_1$, and by then
- successively applying the transformation guaranteed by the restricted statement (8.44) to topmost, yet untreated TRANS_1-applications that originate from TRANS_1-applications in $\tilde{\mathcal{D}}_1$ and in $\tilde{\mathcal{D}}_2$ (i.e. that are “residuals” of TRANS_1-applications in $\tilde{\mathcal{D}}_1$ or $\tilde{\mathcal{D}}_2$ under successive transformations carried out in this step according to the restricted statement of the lemma).

The restricted statement of the lemma can be shown, for all derivations $\mathcal{D}$ in $\mathbf{HB}^{=}_{\mathbf{0}} + {}^*(\mu\!-\!\perp)^{\perp\text{der}}_{l/r} + \text{SYMM} + \{\text{TRANS}_k\}_k$ of the form (8.42) with immediate subderivations $\mathcal{D}_1$ and $\mathcal{D}_2$, by induction on $|\mathcal{D}_1| + |\mathcal{D}_2|$ in an analogous way as in our proof above for Lemma 8.2.15. Namely, by treating the case with an axiom in either premise, and by upwards-permuting, if necessary, the application of TRANS_1 at the bottom of $\mathcal{D}$ over one or over two $\mathbf{HB}^{=}_{\mathbf{0}} + {}^*(\mu\!-\!\perp)^{\perp\text{der}}_{l/r}$-applications immediately above either of its premise in $\mathcal{D}_1$ or/and in $\mathcal{D}_2$. The only additional case to be treated here consists in the situation that $\mathcal{D}_1$ or $\mathcal{D}_2$ or both of them end with an application of TRANS_k, for some $k \in \omega\backslash\{0\}$. Here a desired derivation $\mathcal{D}'$ in $\mathbf{HB}^{=}_{\mathbf{0}} + {}^*(\mu\!-\!\perp)^{\perp\text{der}}_{l/r} + \text{SYMM} + \{\text{TRANS}_k\}_k$ of the form (8.43) with (i)–(iv) and such that $\mathcal{D}'$ mimics $\mathcal{D}$ can be found simply by amalgamating, by $\{\text{TRANS}_k\}_k$-amalgamation steps, the application of TRANS_1 at the bottom of $\mathcal{D}$

with the application(s) of $\{\mathrm{TRANS}_k\}_k$-rules at the bottom of $\mathcal{D}_1$ and/or $\mathcal{D}_2$ (since $\mathcal{D}_1$ and $\mathcal{D}_2$ are already of the required form by the assumption of the restricted statement).

□

Eventually the following lemma generalizes Lemma 8.2.16 into a statement that asserts the existence of a transformation which is applicable to *all* derivations in $\mathbf{HB}^{=}_{\mathbf{0}} + {}^{*}(\mu{-}\bot)^{\bot\mathrm{der}}_{l/r} + \mathrm{SYMM} + \{\mathrm{TRANS}_k\}_k$.

Lemma 8.2.17. *Suppose that $\mathcal{D}$ is a derivation in* $\mathbf{HB}^{=}_{\mathbf{0}} + {}^{*}(\mu{-}\bot)^{\bot der}_{l/r} + \mathrm{SYMM} + {}$ $+ \{\mathrm{TRANS}_k\}_k$ *that fulfills the condition* (8.37) *on its open marked assumptions.*

Then $\mathcal{D}$ can effectively be transformed, by utilizing appropriate ones of the steps (a)–(e) as described in Lemma 8.2.15 and Lemma 8.2.16, into a derivation $\mathcal{D}'$ in $\mathbf{HB}^{=}_{\mathbf{0}} + {}^{*}(\mu{-}\bot)^{\bot der}_{l/r} + \mathrm{SYMM} + \{\mathrm{TRANS}_k\}_k$ *that is of the form* (8.43)*, where (i), (ii), (iii), and (iv) in Lemma 8.2.16 are fulfilled, and such that $\mathcal{D}'$ has the same conclusion and the same open assumption classes as $\mathcal{D}$.*

Sketch of the Proof. In view of Lemma 8.2.14 it suffices to show the restriction of the statement of the lemma to such derivations $\mathcal{D}$ in $\mathbf{HB}^{=}_{\mathbf{0}} + {}^{*}(\mu{-}\bot)^{\bot\mathrm{der}}_{l/r} + \mathrm{SYMM} + {}$ $+ \{\mathrm{TRANS}_k\}_k$ that contain applications of SYMM only immediately below marked assumptions as single applications. And this restricted statement can be shown by induction on the depth $|\mathcal{D}|$ of *such* derivations $\mathcal{D}$ in $\mathbf{HB}^{=}_{\mathbf{0}} + {}^{*}(\mu{-}\bot)^{\bot\mathrm{der}}_{l/r} + \mathrm{SYMM} + {}$ $+ \{\mathrm{TRANS}_k\}_k$; in the induction-step, the cases in which the derivation $\mathcal{D}$ has a bottommost application of SYMM, ${}^{*}(\mu{-}\bot)^{\bot\mathrm{der}}_{l/r}$, or of a rule of $\mathbf{HB}^{=}_{\mathbf{0}}$ can be settled in a straightforward manner; in the case in which $\mathcal{D}$ has a bottommost application of TRANS_k, for some $k \in \omega\backslash\{0\}$, this application is first mimicked by k applications of TRANS_1, by a $(\{\mathrm{TRANS}_k\}_k \rightsquigarrow \mathrm{TRANS}_1)$-mimicking step, and then Lemma 8.2.16 is applied to the bottommost of these TRANS_1-applications. □

For our proof later of Lemma 8.2.13 we will actually need a version of Lemma 8.2.17 in which the 'auxiliary' rules ${}^{*}(\mu{-}\bot)^{\bot\mathrm{der}}_{l/r}$ are not mentioned any more. The next lemma is such a statement, which is an easy consequence of Lemma 8.2.17 in the light of Proposition 8.2.6, and more precisely, in the light of the possibility to eliminate applications of ${}^{*}(\mu{-}\bot)^{\bot\mathrm{der}}_{l/r}$ by $({}^{*}(\mu{-}\bot)^{\bot\mathrm{der}}_{l/r} \rightsquigarrow (\mu{-}\bot)^{\bot\mathrm{der}}_{l/r}, \mathrm{FOLD}_{l/r})$-mimicking steps.

Lemma 8.2.18. *Let $\mathcal{D}$ be a derivation in* $\mathbf{HB}^{=}_{\mathbf{0}} + \mathrm{SYMM} + \{\mathrm{TRANS}_k\}_k$ *that fulfills the condition* (8.37) *on its open marked assumptions.*

Then $\mathcal{D}$ can effectively be transformed, by appropriate ones of the steps (a)–(e) as described in Lemma 8.2.15 and Lemma 8.2.16, and additionally by

(f) $({}^{*}(\mu{-}\bot)^{\bot der}_{l/r} \rightsquigarrow (\mu{-}\bot)^{\bot der}_{l/r}, \mathrm{FOLD}_{l/r})$*-mimicking steps,*

into a derivation $\mathcal{D}'$ in $\mathbf{HB}^{=}_{\mathbf{0}} + \mathrm{SYMM} + \{\mathrm{TRANS}_k\}_k$ *that is of the form* (8.43) *where the four conditions*

(i) $I_1, I_2 \subseteq \omega$*,* $I_1 \cap I_2 = \emptyset$*, and* $I_1 \cup I_2 = \{1, \ldots, m\}$ *for some* $m \in \omega$ *(if* $m = 0$ *then* $I_1 = I_2 = \emptyset$*),*

(ii) *for all* $i \in I_1$, $k_i \in \omega\backslash\{0\}$, $\tau_{i1}, \tau_{i2}, \sigma_{i1}, \sigma_{i2} \in \mu Tp$, *and* $\mathcal{D}_{i1}$, $\mathcal{D}_{i2}$, $\ldots$, $\mathcal{D}_{i(k_i+1)}$ *are derivations in* $\mathbf{HB}_{\mathbf{0}}^{=} + \mathrm{SYMM} + \{\mathrm{TRANS}_k\}_k$, *and furthermore, for all* $i \in I_2$, u_i *is an assumption marker and* $\tau_{i1}, \tau_{i2}, \sigma_{i1}, \sigma_{i2} \in \mu Tp$,

(iii) $\mathcal{DC} \in \mathcal{D}er\mathcal{C}txt_m(\mathbf{HB}_{\mathbf{0}}^{=})$ *is the* $\mathbf{HB}_{\mathbf{0}}^{=}$*-end-derivation-context* $\lceil \mathcal{D} \rceil_{\mathbf{HB}_{\mathbf{0}}^{=}}^{\{[\,]_i\}_i}$ *of* $\mathcal{D}$,

(iv) *all derivations* $\mathcal{D}_{i1}$, $\mathcal{D}_{i2}$, $\ldots$, $\mathcal{D}_{i(k_i+1)}$, *for* $i \in I_1$, *are such derivations in* $\mathbf{HB}_{\mathbf{0}}^{=} + \mathrm{SYMM} + \{\mathrm{TRANS}_k\}_k$ *that are of one of the four possible forms in* (8.39) *for respective* $\tilde{\tau}_1, \tilde{\tau}_2, \tilde{\sigma}_1, \tilde{\sigma}_2 \in \mu Tp$ *and assumption markers* $\tilde{u}$, *and where not all of* $\mathcal{D}_{i1}$, $\mathcal{D}_{i2}$, $\ldots$, $\mathcal{D}_{i(k_i+1)}$ *end with an application of* ARROW,

are fulfilled, and such that $\mathcal{D}'$ *has the same conclusion and the same open assumption classes as* $\mathcal{D}$.

Proof. The assertion of the lemma is an immediate consequence of Lemma 8.2.17 in view of the following easily verifiable fact: every derivation $\tilde{\mathcal{D}}$ in the system $\mathbf{HB}_{\mathbf{0}}^{=} + {}^{*}(\mu - \bot)_{l/r}^{\bot\mathrm{der}} + \mathrm{SYMM} + \{\mathrm{TRANS}_k\}_k$ that is of the form as described in the conclusion of the statement of Lemma 8.2.16 (referring to a prooftree of the form (8.43) and the conditions (i)–(iv) in Lemma 8.2.16) can effectively be transformed, by eliminating all applications of rules ${}^{*}(\mu - \bot)_{l/r}^{\bot\mathrm{der}}$ using (${}^{*}(\mu - \bot)_{l/r}^{\bot\mathrm{der}} \rightsquigarrow$ $\rightsquigarrow (\mu - \bot)_{l/r}^{\bot\mathrm{der}}$, $\mathrm{FOLD}_{l/r}$)-mimicking steps, into a mimicking derivation $\mathcal{D}'$ for $\tilde{\mathcal{D}}$ in $\mathbf{HB}_{\mathbf{0}}^{=} + \mathrm{SYMM} + \{\mathrm{TRANS}_k\}_k$ of the form (8.43) such that the conditions (i)–(iv) in the lemma to prove here are fulfilled. □

In derivations in $\mathbf{HB}_{\mathbf{0}}^{=} + \mathrm{SYMM} + \{\mathrm{TRANS}_k\}_k$ of a form (8.43) with the conditions (i)–(iv) in Lemma 8.2.18 on the occurring designations fulfilled the displayed occurrences of applications of SYMM and of TRANS_{k_i} are prevented from being permuted further upwards by the presence, just above them, of marked assumptions, or of applications of ARROW/FIX. A possibility to proceed in this situation consists in 'unfolding' a derivation $\mathcal{D}$ of this form with respect to bindings of assumptions to applications of ARROW/FIX, in such a way, that it then becomes possible again to permute the applications of SYMM and of TRANS_{k_i} immediately above the $\mathbf{HB}_{\mathbf{0}}^{=}$-end-derivation-context of $\mathcal{D}$ still further upwards.

For this purpose we use two kinds of transformations for 'unfolding' derivations in $\mathbf{HB}_{\mathbf{0}}^{=} + \mathrm{SYMM} + \{\mathrm{TRANS}_k\}_k$ with respect to bindings in them of marked assumptions to respective applications of ARROW/FIX where these assumptions are discharged: firstly, the transformation illustrated in Figure 8.5 of 'unfolding' a derivation above a the conclusion of a particular occurrence of an ARROW/FIX-application, and secondly, the transformation described in Figure 8.6 of 'unfolding' a derivation above a particular occurrence of a discharged assumption. In both cases the goal for performing a transformation of this kind to a derivation $\mathcal{D}$ in $\mathbf{HB}_{\mathbf{0}}^{=} + \mathrm{SYMM} + \{\mathrm{TRANS}_k\}_k$ consists in extending $\mathcal{D}$ above a considered occurrence of a marked assumption or of a conclusion of an ARROW/FIX application *in such a way* that, firstly, a mimicking derivation $\tilde{\mathcal{D}}$ for $\mathcal{D}$ is effectively produced, and that secondly, the considered formula occurrence becomes the conclusion of an application of ARROW in the transformed derivation $\tilde{\mathcal{D}}$.

Figure 8.5: ‘Unfolding’ of a derivation $\mathcal{D}$ in $\mathbf{HB}_{\mathbf{0}}^= + \mathrm{SYMM} + \{\mathrm{TRANS}_k\}_k$ above a particular occurrence of a conclusion of an ARROW/FIX-application (this occurrence in the original derivation $\mathcal{D}$ on the right-hand side, and the corresponding formula occurrence in the resulting derivation $\tilde{\mathcal{D}}$ on the left-hand side are both typeset in boldface).

$$
\begin{array}{c}
[\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2]^u \\
\mathcal{D}_a \\
\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \ \ \scriptstyle{\mathrm{ARROW/FIX},\ u} \\
\boldsymbol{(\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2)} \\
\mathcal{DC}_0 \\
\tau = \sigma
\end{array}
\quad \longmapsto \quad
\begin{array}{c}
[\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2]^u \\
\mathcal{D}_a \\
\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \ \ \scriptstyle{\mathrm{ARROW/FIX},\ u} \\
[\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2] \\
\mathcal{D}'_a \\
\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \ \ \scriptstyle{\mathrm{ARROW}} \\
\boldsymbol{(\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2)} \\
\mathcal{DC}_0 \\
\tau = \sigma
\end{array}
$$

‘Unfolding’ of a derivation above a conclusion of ARROW/FIX: in a transformation step of the form shown in Figure 8.5, the displayed assumption class $[\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2]^u$ at the top of the symbolic prooftree for a derivation $\mathcal{D}$ at the left-hand side is assumed to be discharged at the displayed application of ARROW/FIX; and the same holds, respectively, for the assumption class of the form $[\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2]^u$ that is displayed at the top of the symbolic prooftree of the transformed derivation $\tilde{\mathcal{D}}$ on the right-hand side. The subderivation $\mathcal{D}'_a$ in $\tilde{\mathcal{D}}$ arises from the subderivation $\mathcal{D}_a$ of $\mathcal{D}$ by changing the rule label of the bottommost application of $\mathcal{D}_a$ from ARROW/FIX to ARROW with resulting derivation $\mathcal{D}_a^{(0)}$, and by taking an appropriate “variant-derivation” of $\mathcal{D}_a^{(0)}$ such that unwanted bindings of assumptions due to the construction of $\tilde{\mathcal{D}}$ from $\mathcal{D}$ are avoided; more precisely, $\mathcal{D}'_a$ results from $\mathcal{D}_a^{(0)}$ by performing renamings of assumption markers different from u of discharged assumptions in $\mathcal{D}_a$ *in such a way* that $\mathcal{D}'_a$ mimics $\mathcal{D}_a^{(0)}$ and such that no open assumption in a new copy of $\mathcal{D}_a$ at the top of $\tilde{\mathcal{D}}$ gets discharged in the part $\mathcal{D}'_a$ of $\tilde{\mathcal{D}}$. In a transformation step shown in Figure 8.5, the desired occurrence of a conclusion of an ARROW/FIX-application, above which the derivation on the left-hand side is ‘unfolded’, and the corresponding formula occurrence in the transformed derivation $\tilde{\mathcal{D}}$ on the right-hand side are both typeset in boldface.

‘Unfolding’ of a derivation above a discharged assumption: a transformation step shown in Figure 8.6 of ‘unfolding’ a derivation $\mathcal{D}$ in $\mathbf{HB}_{\mathbf{0}}^= + \mathrm{SYMM} + \{\mathrm{TRANS}_k\}_k$ above the occurrence, typeset in boldface, of a marked assumption in $\mathcal{D}$ has to be understood in a similar way, but the stipulations used there are slightly more involved. In particular, in a derivation $\mathcal{D}$ that is depicted on the right-hand side of such a step there is only one particular occurrence of a marked assumption $(\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2)^u$ considered from the assumption class $[\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2]^u$ that is assumed to be discharged at the displayed application of ARROW/FIX at

Figure 8.6: 'Unfolding' of a derivation $\mathcal{D}$ in $\mathbf{HB}_{\mathbf{0}}^{=} + \text{SYMM} + \{\text{TRANS}_k\}_k$ above a particular occurrence of a discharged assumption (this assumption occurrence in the original derivation $\mathcal{D}$ on the left-hand side, and the corresponding formula occurrence in the resulting derivation $\tilde{\mathcal{D}}$ on the right-hand side are both typeset in boldface).

$$\begin{array}{c} [(\boldsymbol{\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2})]^u \\ \mathcal{D}_a \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots \;\text{ARROW/FIX, } u \\ (\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2) \\ \mathcal{DC}_0 \\ \tau = \sigma \end{array} \quad\longmapsto\quad \begin{array}{c} [\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2]^u \\ \mathcal{D}_a \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots \;\text{ARROW/FIX, } u \\ [\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2] \\ \mathcal{D}''_a \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \;\text{ARROW} \\ (\boldsymbol{\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2}) \\ \mathcal{D}'_a \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots \;\begin{array}{c}\text{ARROW/FIX, } u \\ \text{or ARROW}\end{array} \\ (\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2) \\ \mathcal{DC}_0 \\ \tau = \sigma \end{array}$$

the bottom of the subderivation $\mathcal{D}_a$. The derivation $\mathcal{D}''_a$ in the symbolic prooftree for the resulting derivation $\tilde{\mathcal{D}}$ on the right-hand side of the transformation step arises from $\mathcal{D}_a$ by changing the label for its bottommost application from ARROW/FIX to ARROW with resulting derivation $\mathcal{D}_a^{(0)}$, and by making sure, through renamings of assumption markers of discharged assumption classes in $\mathcal{D}_a^{(0)}$, that no open assumptions of a new copy of $\mathcal{D}_a$ at the top of $\tilde{\mathcal{D}}$ are discharged in the part $\mathcal{D}''_a$ of $\tilde{\mathcal{D}}$. And the subderivation $\mathcal{D}'_a$ of $\mathcal{D}_a$ arises by similar appropriate renamings of assumption markers in the derivation $\mathcal{D}_a^{(00)}$, which results from $\mathcal{D}_a$ by changing the rule label of its last rule application from ARROW/FIX to ARROW in case that the considered occurrence of $(\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2)^u$ at the top of $\mathcal{D}$ is the only member of the open assumption class displayed there, and which is equal to $\mathcal{D}_a$ otherwise. And furthermore, the (respective) assumption classes $[\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2]^u$ in copies of $\mathcal{D}_a$ at the top of $\tilde{\mathcal{D}}$ are assumed to be discharged respectively at displayed occurrences of ARROW/FIX at the bottom of displayed copies of $\mathcal{D}_a$ in the symbolic prooftree for $\tilde{\mathcal{D}}$; contrasting with this, the bottommost displayed application of (ARROW/FIX, u) in $\tilde{\mathcal{D}}$ is assumed to discharge precisely all those marked assumptions $[\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2]^u$ that are "untouched" by 'unfolding' $\mathcal{D}$ above the considered occurrence of $[\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2]^u$ in $\mathcal{D}$. The corresponding formula occurrence in the transformed derivation $\tilde{\mathcal{D}}$ of the assumption above which $\mathcal{D}$ was extended is also typeset in boldface in the derivation $\tilde{\mathcal{D}}$ on the right-hand side.

The subsequent lemma will later be used in the proof of Lemma 8.2.13 to justify the induction step in the inductive definition, for arbitrary derivations $\mathcal{D}$ in $\mathbf{HB}_{\mathbf{0}}^{=} +$ $+ \text{SYMM} + \{\text{TRANS}_k\}_k$, of a sequence $\mathcal{SD} = \langle \mathcal{D}^{(n)} \rangle_{n \in I}$ of derivations in $\mathbf{HB}_{\mathbf{0}}^{=} +$

$+\,\mathrm{SYMM}+\{\mathrm{TRANS}_k\}_k$ with the properties (i)–(vi) in Lemma 8.2.13. This lemma states that every derivation $\mathcal{D}$ of the special form as described in the consequence of Lemma 8.2.18 can be transformed into a mimicking derivation $\mathcal{D}'$ of the same special form, but with a strictly larger $\mathbf{HB}^{=}_{\mathbf{0}}$-end-derivation-context. The transformation used hereby proceeds essentially by applying the 'unfolding'-operations just explained, by performing upwards-permutation steps for applications of rules from $\{\mathrm{TRANS}_k\}_k$, and furthermore by utilizing the transformations (a)–(f) mentioned in Lemma 8.2.18.

Lemma 8.2.19. *Let $\tau, \sigma \in \mu Tp$. Furthermore, let $\mathcal{D}$ be a derivation in $\mathbf{HB}^{=}_{\mathbf{0}}+$ $+\,\mathrm{SYMM}+\{\mathrm{TRANS}_k\}_k$*

(I) without open assumption classes and with conclusion $\tau = \sigma$,

(II) that contains at least one application of SYMM *or* TRANS_k*, for some $k \in \omega$,*

(III) that does not contain successive applications of REN *nor nlμb-decreasing applications of* $\mathrm{FOLD}_{l/r}$*,*

(IV) that is of the form as described in the consequence of Lemma 8.2.18; in particular, $\mathcal{D}$ is of the form (8.43) *such that (i), (ii), (iii), and (iv) in Lemma 8.2.18 hold, in particular with $\mathbf{HB}^{=}_{\mathbf{0}}$-end-derivation-context $\mathcal{DC} \in \mathcal{D}er\mathcal{C}txt_m(\mathbf{HB}^{=}_{\mathbf{0}})$, for some $m \in \omega \backslash \{0\}$.*

Then $\mathcal{D}$ can be transformed effectively, by the steps (a)–(f) described or referred to in Lemma 8.2.18, and additionally by

(g) 'unfolding'-operations of $\mathbf{HB}^{=}_{\mathbf{0}}+\mathrm{SYMM}+\{\mathrm{TRANS}_k\}_k$-derivations that are defined with the help of the illustrations in Figure 8.5 and Figure 8.6,

(h) upwards-permutation steps of $\{\mathrm{TRANS}_k\}_k$-applications over ARROW*-applications as described in Proposition 8.2.20 below,*

(i) elimination steps for nlμb-decreasing $\mathrm{FOLD}_{l/r}$*-applications as described in the proof of item (iii) of Lemma C.10, Appendix C,*

into a derivation $\mathcal{D}'$ in $\mathbf{HB}^{=}_{\mathbf{0}}+\mathrm{SYMM}+\{\mathrm{TRANS}_k\}_k$

(A) without open assumption classes and with conclusion $\tau = \sigma$, and

(B) that does not contain successive applications of REN *nor nlμb-decreasing applications of* $\mathrm{FOLD}_{l/r}$*,*

(C) that is also of the form as described in the consequence of Lemma 8.2.18, with a $\mathbf{HB}^{=}_{\mathbf{0}}$-end-part $\mathcal{DC}' \in \mathcal{D}er\mathcal{C}txt_{m'}(\mathbf{HB}^{=}_{\mathbf{0}})$, for some $m' \in \omega$, where $\mathcal{DC}'$ arises from $\mathcal{DC}$ by hole-filling of the occurrences of $[\,]_1, \ldots, [\,]_m$ with appropriate derivation-contexts $\mathcal{DC}_1, \mathcal{DC}_2, \ldots, \mathcal{DC}_m$; furthermore

$$s(\mathcal{DC}') > s(\mathcal{DC}) \tag{8.45}$$

holds, i.e. $\mathcal{DC}'$ *is of greater size than* $\mathcal{DC}$*; and if* $\mathcal{D}'$ *still contains applications of* SYMM *or* $\{\text{TRANS}_k\}_k$*, then also*

$$h_{\mathbf{HB}_{\mathbf{0}}^{=}}(\mathcal{D}') > h_{\mathbf{HB}_{\mathbf{0}}^{=}}(\mathcal{D}) \tag{8.46}$$

holds, i.e. the $\mathbf{HB}_{\mathbf{0}}^{=}$*-height of* $\mathcal{D}'$ *is greater than that of* $\mathcal{D}$*.*

We *prove* this lemma immediately below the following proposition, which states and describes the possibility to permute an arbitrary $\{\text{TRANS}_k\}_k$-application upwards in a derivation of $\mathbf{HB}_{\mathbf{0}}^{=} + \text{SYMM} + \{\text{TRANS}_k\}_k$ provided that all immediate subderivations end with an application of ARROW.

Proposition 8.2.20 (Upwards-permutation of $\{\text{TRANS}_k\}_k$-applications over ARROW-applications). *Let* $\mathcal{D}$ *be a derivation in* $\mathbf{HB}_{\mathbf{0}}^{=} + \text{SYMM} + \{\text{TRANS}_k\}_k$ *that ends with an application of* TRANS_k*, for some* $k \in \omega\backslash\{0\}$*, and that is such that all immediate subderivations of* $\mathcal{D}$ *end with an application of* ARROW*. Then the application of* TRANS_k *at the bottom of* $\mathcal{D}$ *can be permuted upwards simultaneously over all applications of* ARROW *immediately above this* TRANS_k*-application.*

More precisely, every derivation $\mathcal{D}$ *in* $\mathbf{HB}_{\mathbf{0}}^{=} + \text{SYMM} + \{\text{TRANS}_k\}_k$ *of the form*

$$\dfrac{\dfrac{\begin{matrix}\mathcal{D}_{11}\\ \tau_1' = \rho_{11}\end{matrix} \qquad \begin{matrix}\mathcal{D}_{12}\\ \tau_2' = \rho_{12}\end{matrix}}{\tau_1' \to \tau_2' = \rho_{11} \to \rho_{12}}\,\text{ARROW} \qquad \dots \qquad \dfrac{\begin{matrix}\mathcal{D}_{(k+1)1}\\ \rho_{k1}' = \sigma_1'\end{matrix} \qquad \begin{matrix}\mathcal{D}_{(k+1)2}\\ \rho_{k2}' = \sigma_2'\end{matrix}}{\rho_{k1}' \to \rho_{k2}' = \sigma_1' \to \sigma_2'}\,\text{ARROW}}{\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2}\,\text{TRANS}_k$$

can be transformed into a derivation $\mathcal{D}'$ *in* $\mathbf{HB}_{\mathbf{0}}^{=} + \text{SYMM} + \{\text{TRANS}_k\}_k$ *of the form*

$$\dfrac{\dfrac{\begin{matrix}\mathcal{D}_{11}\\ \tau_1' = \rho_{11}\end{matrix} \quad \dots \quad \begin{matrix}\mathcal{D}_{(k+1)1}\\ \rho_{k1}' = \sigma_1'\end{matrix}}{\tau_1 = \sigma_1}\,\text{TRANS}_k \qquad \dfrac{\begin{matrix}\mathcal{D}_{12}\\ \tau_2' = \rho_{12}\end{matrix} \quad \dots \quad \begin{matrix}\mathcal{D}_{(k+1)2}\\ \rho_{k2}' = \sigma_2'\end{matrix}}{\tau_2 = \sigma_2}\,\text{TRANS}_k}{\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2}\,\text{ARROW}$$

with the same conclusion and with the same open assumption classes as $\mathcal{D}$*.*

Proof of Lemma 8.2.19. Let $\tau, \sigma \in \mu Tp$ be arbitrary, and let $\mathcal{D}$ be an arbitrary derivation in $\mathbf{HB}_{\mathbf{0}}^{=} + \text{SYMM} + \{\text{TRANS}_k\}_k$ such that the conditions (I)–(IV) in the lemma are met. In particular, $\mathcal{D}$ has the conclusion $\tau = \sigma$, does not contain open assumptions, contains at least one application of SYMM or TRANS_k, for some $k \in \omega\backslash\{0\}$, does not contain successive applications of REN nor nlμb-decreasing applications of $\mathrm{FOLD}_{l/r}$, and is of the form

$$\begin{array}{c}
\mathrm{TRANS}_{k_i}\;\dfrac{\mathcal{D}_{i1}\quad \mathcal{D}_{i2}\quad \ldots\quad \mathcal{D}_{i(k_i+1)}}{\{\;(\,\tau_{i1}\to\tau_{i2}=\sigma_{i1}\to\sigma_{i2}\,)_i\;\}_{i\in I_1}}
\qquad
\dfrac{\overset{(\text{Assm})}{(\sigma_{i1}\to\sigma_{i2}=\tau_{i1}\to\tau_{i2})^{u_i}}}{\{\;(\,\tau_{i1}\to\tau_{i2}=\sigma_{i1}\to\sigma_{i2}\,)_i\;\}_{i\in I_2}}\;\mathrm{SYMM}\\
\mathcal{DC}\\
\tau=\sigma
\end{array}\tag{8.47}$$

where the conditions (i)–(iv) in Lemma 8.2.18 are fulfilled on the occurring designations, and where in particular $\mathcal{DC}\in\mathcal{D}er\mathcal{C}txt_m(\mathbf{HB}^{=}_{\mathbf{0}})$, for some $m\in\omega\backslash\{0\}$, is the $\mathbf{HB}^{=}_{\mathbf{0}}$-end-derivation-context of $\mathcal{D}$.

By 'unfolding' the derivation $\mathcal{D}$ immediately above the bottommost of its SYMM- and $\{\mathrm{TRANS}_k\}_k$-applications, which are displayed schematically in (8.47), i.e. by applying the transformations described in Figure 8.5 and Figure 8.6 above all such occurrences of marked assumptions or of conclusions of ARROW/FIX that are immediately, or after a single SYMM-application, followed by one of the occurrences of SYMM or TRANS_{k_i} displayed in (8.47), and by, if necessary, permuting single occurrences of SYMM-applications just above displayed occurrences of TRANS_{k_i} upwards over newly arising occurrences of ARROW, a derivation $\tilde{\mathcal{D}}$ in $\mathbf{HB}^{=}_{\mathbf{0}}+$ $+\,\mathrm{SYMM}+\{\mathrm{TRANS}_k\}_k$ of the form

$$\begin{array}{c}
\mathrm{TRANS}_{k_i}\;\dfrac{
\mathrm{ARROW}\;\begin{array}{c}\tilde{\mathcal{D}}_{i1}\\ \cdots\cdots\cdots\cdots\cdots\cdots\\ (\tau_{i1}\to\tau_{i2}=\rho_{i11}\to\rho_{i12})\end{array}
\quad\ldots\quad
\begin{array}{c}\tilde{\mathcal{D}}_{i2}\\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\\ (\rho'_{i(k_i+1)1}\to\rho'_{i(k_i+1)2}=\sigma_{i1}\to\sigma_{i2})\end{array}\;\mathrm{ARROW}
}{\{\;(\,\tau_{i1}\to\tau_{i2}=\sigma_{i1}\to\sigma_{i2}\,)_i\;\}_{i\in I_1}}
\qquad
\dfrac{\begin{array}{c}\tilde{\mathcal{D}}_{i1}\\ \cdots\cdots\cdots\cdots\cdots\cdots\\ (\sigma_{i1}\to\sigma_{i2}=\tau_{i1}\to\tau_{i2})\end{array}\;\mathrm{ARROW}}{\{\;(\,\tau_{i1}\to\tau_{i2}=\sigma_{i1}\to\sigma_{i2}\,)_i\;\}_{i\in I_2}}\;\mathrm{SYMM}\\
\mathcal{DC}\\
\tau=\sigma
\end{array}\tag{8.48}$$

without open assumptions can be found. The mentioned 'unfolding'-operations have here been possible due to the fact that $\mathcal{D}$ does not contain open assumption classes. In $\tilde{\mathcal{D}}$ all bottommost applications of SYMM and of rules $\{\mathrm{TRANS}_k\}_k$ now have exclusively such immediate subderivations that end with an application of ARROW.

The $\mathbf{HB}^{=}_{\mathbf{0}}$-end-derivation-context $\mathcal{DC}$ of $\tilde{\mathcal{D}}$ (and also of $\mathcal{D}$) can now be enlarged by permuting the bottommost applications of TRANS_{k_i}, for $i\in I_1$, and SYMM, which are displayed in (8.48), upwards over the preceding rule applications of ARROW. Upwards-permutation of SYMM- over ARROW-applications is an easy matter, while upwards-permutation of applications of $\{\mathrm{TRANS}_k\}_k$ over applications of ARROW is described in Proposition 8.2.20 (but it is clearly also straightforward). By performing these permutations of rule applications, we arrive here at a derivation

$\tilde{\tilde{\mathcal{D}}}$ in $\mathbf{HB}^{=}_{\mathbf{0}} + \text{SYMM} + \{\text{TRANS}_k\}_k$ of the form

$$\begin{array}{c}
\left\{\ \text{ARROW}\ \dfrac{\begin{array}{cc} \text{TRANS}_{k_i}\ \overset{\tilde{\tilde{\mathcal{D}}}_{i1}}{\cdots\cdots\cdots} & \overset{\tilde{\tilde{\mathcal{D}}}_{i2}}{\cdots\cdots\cdots}\ \text{TRANS}_{k_i} \\ \tau_{i1}=\sigma_{i1} & \tau_{i2}=\sigma_{i2} \end{array}}{(\,\tau_{i1}\to\tau_{i2}=\sigma_{i1}\to\sigma_{i2}\,)_i}\ \right\}_{i\in I_1}
\quad
\left\{\ \dfrac{\begin{array}{cc} \text{SYMM}\ \overset{\tilde{\tilde{\mathcal{D}}}_{i1}}{\cdots\cdots\cdots} & \overset{\tilde{\tilde{\mathcal{D}}}_{i2}}{\cdots\cdots\cdots}\ \text{SYMM} \\ \tau_{i1}=\sigma_{i1} & \tau_{i2}=\sigma_{i2} \end{array}}{(\,\tau_{i1}\to\tau_{i2}=\sigma_{i1}\to\sigma_{i2}\,)_i}\ \text{ARROW}\ \right\}_{i\in I_2} \\
\mathcal{DC} \\
\tau=\sigma
\end{array}
\tag{8.49}$$

without open assumption-classes. $\tilde{\tilde{\mathcal{D}}}$ obviously has a $\mathbf{HB}^{=}_{\mathbf{0}}$-end-derivation-context of greater size than $\tilde{\mathcal{D}}$, and also the $\mathbf{HB}^{=}_{\mathbf{0}}$-height of $\tilde{\tilde{\mathcal{D}}}$ is now strictly greater than that of $\mathcal{D}$ (note that $\tilde{\tilde{\mathcal{D}}}$ still contains applications of SYMM or of rules from $\{\text{TRANS}_k\}_k$). However, the derivation $\tilde{\tilde{\mathcal{D}}}$ is not yet again of a form as described in the consequence of Lemma 8.2.18 (in particular, the immediate subderivations of the derivations $\tilde{\tilde{\mathcal{D}}}_{i1}$ and $\tilde{\tilde{\mathcal{D}}}_{i2}$, for $i \in I_1 \cup I_2$, may not yet be of one of the four possible forms in (8.43)).

By applying the transformation stated by Lemma 8.2.18, for all $i \in I_1 \cup I_2$, to each of the subderivations $\tilde{\tilde{\mathcal{D}}}_{i1}$ and $\tilde{\tilde{\mathcal{D}}}_{i2}$ of $\tilde{\tilde{\mathcal{D}}}$, mimicking derivations $\mathcal{D}'_{i1}$ and $\mathcal{D}'_{i2}$ in $\mathbf{HB}^{=}_{\mathbf{0}} + \text{SYMM} + \{\text{TRANS}_k\}_k$ for $\tilde{\tilde{\mathcal{D}}}_{i1}$ and $\tilde{\tilde{\mathcal{D}}}_{i2}$ can effectively be found that are of the form as described in the consequence of Lemma 8.2.18; in particular, for all $i \in I_1 \cup I_2$, $\mathcal{D}'_{i1}$ and $\mathcal{D}'_{i2}$ are of the respective forms

$$\begin{array}{c} \{\,(\mathcal{D}'_{i1j})_j\,\}_{j\in J_{i1}} \\ \mathcal{DC}'_{i1} \\ \tau_{i1}=\sigma_{i1} \end{array}
\qquad\qquad
\begin{array}{c} \{\,(\mathcal{D}'_{i2j})_j\,\}_{j\in J_{i2}} \\ \mathcal{DC}'_{i2} \\ \tau_{i2}=\sigma_{i2} \end{array}$$

where $J_{i1}, J_{i2} \subseteq \omega$ are finite index sets, $\mathcal{DC}'_{i1}$ and $\mathcal{DC}'_{i2}$ are the $\mathbf{HB}^{=}_{\mathbf{0}}$-end-derivation contexts of $\mathcal{D}'_{i1}$ and $\mathcal{D}'_{i2}$ and where each derivation $\mathcal{D}'_{i1j}$, for $j \in J_{i1}$, and $\mathcal{D}'_{i2j}$, for $j \in J_{i2}$, ends with an application of SYMM or a rule in $\{\text{TRANS}_k\}_k$. In the following we assume that all derivations $\mathcal{D}_{ikj}$, for $i \in I_1 \cup I_2$, $k \in \{1,2\}$ and $j \in J_{ik}$, do not contain successive applications of REN nor nlμb-decreasing applications of $\text{FOLD}_{l/r}$; if for some derivation $\mathcal{D}_{ikj}$ this is not the case from the outset, then these applications are respectively removed first (nlμb-decreasing $\text{FOLD}_{l/r}$-applications can be eliminated due to the transformation described in the proof of item (iii) of Lemma C.10, Appendix C). We find now that the derivation $\mathcal{D}'$ in $\mathbf{HB}^{=}_{\mathbf{0}} + {}$ $+ \text{SYMM} + \{\text{TRANS}_k\}_k$ of the form

$$\begin{array}{c}
\left\{\ \dfrac{\begin{array}{cc} \{\,(\mathcal{D}'_{i1j})_j\,\}_{j\in J_{i1}} & \{\,(\mathcal{D}'_{i2j})_j\,\}_{j\in J_{i2}} \\ \mathcal{DC}'_{i1} & \mathcal{DC}'_{i2} \\ \tau_{i1}=\sigma_{i1} & \tau_{i2}=\sigma_{i2} \end{array}}{(\,\tau_{i1}\to\tau_{i2}=\sigma_{i1}\to\sigma_{i2}\,)_i}\ \text{ARROW}\ \right\}_{i\in I_1\cup I_2} \\
\mathcal{DC} \\
\tau=\sigma
\end{array}
\tag{8.50}$$

is a mimicking derivation for $\tilde{\tilde{\mathcal{D}}}$ (and henceforth also for $\tilde{\mathcal{D}}$ and $\mathcal{D}$) and it has the form as described in the consequence of Lemma 8.2.18. Due to the construction of $\mathcal{D}'$ and, in particular, due to the choice of the subderivations $\mathcal{D}'_{ikj}$, the derivation $\mathcal{D}'$ contains neither successive REN-applications nor nlμb-decreasing applications of $\text{FOLD}_{l/r}$. Therefore $\mathcal{D}'$ certainly fulfills the conditions (A) and (B) in the consequence of the lemma. Let now $\mathcal{DC}'$ be the $\mathbf{HB}_{\mathbf{0}}^{=}$-end-derivation-context of $\mathcal{D}'$. Due to the form of $\mathcal{D}'$, the derivation-context $\mathcal{DC}' \in \mathcal{D}er\mathcal{C}txt_{m'}(\mathbf{HB}_{\mathbf{0}}^{=})$ for some $m' \in \omega$, $\mathcal{DC}'$ results from filling the context-holes $[\,]_1, \ldots, [\,]_m$ in $\mathcal{DC}$ by the contexts

$$\frac{\begin{array}{c}(\mathcal{DC}'_{i1})' \\ \tau_{i1} = \sigma_{i1}\end{array} \qquad \begin{array}{c}(\mathcal{DC}'_{i2})' \\ \tau_{i2} = \sigma_{i2}\end{array}}{\tau_{i1} \to \tau_{i2} = \sigma_{i1} \to \sigma_{i2}}\ \text{ARROW}$$

for $i \in I_1 \cup I_2 = \{1, \ldots m\}$, where the derivation-contexts $(\mathcal{DC}'_{i1})'$ and $(\mathcal{DC}'_{i2})'$ arise by exchanging the context-holes in $\mathcal{DC}'_{i1}$ and $\mathcal{DC}'_{i2}$ by respective ones carrying appropriate numbers such that the context-holes $[\,]_1, \ldots, [\,]_{m'}$ in $\mathcal{DC}'$ are ordered in a traversal from let to right as required for a $\mathbf{HB}_{\mathbf{0}}^{=}$-end-derivation-context according to Definition 8.2.11. Furthermore it follows that (8.45) holds, and, if $\mathcal{D}'$ again contains applications of SYMM or $\{\text{TRANS}_k\}_k$, also (8.46). Hence $\mathcal{D}'$ satisfies also the condition (C) in the consequence of the lemma.

In this way we have effectively found, for the derivation $\mathcal{D}$ that we have assumed arbitrarily in $\mathbf{HB}_{\mathbf{0}}^{=} + \text{SYMM} + \{\text{TRANS}_k\}_k$ according to the hypotheses (I)–(IV) of the lemma, a mimicking derivation $\mathcal{D}'$ in the same system that fulfills the properties (A), (B), and (C) in the assertion of the lemma; and furthermore it is easy to check (by going through this proof and collecting the kind of transformation steps that have been performed here, either explicitly or as part of transformations guaranteed by the lemmas we have used) that we have not used other transformation steps that those that are mentioned as the items (a)–(i) in the lemma.

□

Now we have eventually gathered all lemmas that are needed for our proof below of Lemma 8.2.13, the statement that will guarantee the second step (St2)′ in the proof of the theorem of this section, Theorem 8.2.2.

Proof of Lemma 8.2.13. Let $\tau, \sigma \in \mu Tp$ be arbitrary, and let $\mathcal{D}$ be an arbitrary derivation in $\mathbf{HB}_{\mathbf{0}}^{=} + \text{SYMM} + \{\text{TRANS}_k\}_k$ without open assumptions and with conclusion $\tau = \sigma$. We have to show that a sequence $\mathcal{SD} = \langle \mathcal{D}^{(n)} \rangle_{n \in I}$ of derivations in $\mathbf{HB}_{\mathbf{0}}^{=} + \text{SYMM} + \{\text{TRANS}_k\}_k$ can effectively be built, where I is either ω or of the form $[0, n_{\max}] \cup \omega$ for some $n_{\max} \in \omega$, such that $\mathcal{SD}$ fulfills the conditions (i)–(vi) in the lemma.

We proceed by defining the sequence $\mathcal{SD} = \langle \mathcal{D}^{(n)} \rangle_{n \in I}$ inductively. In the induction step we will assume that we have already constructed a finite sequence $\mathcal{SD}^{(n)} = \langle \mathcal{D}^{(0)}, \ldots, \mathcal{D}^{(n)} \rangle$ of derivations in $\mathbf{HB}_{\mathbf{0}}^{=} + \text{SYMM} + \{\text{TRANS}_k\}_k$ with the seven properties

(i)′ $\mathcal{D}^{(0)}$, the first derivation in $\mathcal{SD}^{(0)}$, is the derivation $\mathcal{D}$, (ii)′ all derivations in $\mathcal{SD}^{(n)}$ have conclusion $\tau = \sigma$ and do not contain open assumptions,

(iii)′ all derivations in $\mathcal{SD}^{(n)}$ except $\mathcal{D}^{(0)}$ do not contain successive applications of REN, nor nlμb-decreasing applications of $\mathrm{FOLD}_{l/r}$,

(iv)′ for all $i \in \{0, \ldots, n-1\}$, the derivation $\mathcal{D}^{(i+1)}$ is the result of applying an effective proof-theoretic transformation to the derivation $\mathcal{D}^{(i)}$,

(v)′ all derivations in $\mathcal{SD}^{(n)}$ contain applications of SYMM or of rules from the family $\{\mathrm{TRANS}_k\}_k$,

(vi)′ all derivations in $\mathcal{SD}^{(n)}$ except $\mathcal{D}^{(0)}$ are of a form (8.43) such that the conditions (i)–(iv) in Lemma 8.2.16 on the occurring designations are fulfilled,

(vii)′ the $\mathbf{HB}_{\mathbf{0}}^{=}$-heights of the derivations in the sequence $\mathcal{SD}$ are strictly increasing after the first step, i.e. it holds:

$$h_{\mathbf{HB}_{\mathbf{0}}^{=}}(\mathcal{D}^{(n)}) > h_{\mathbf{HB}_{\mathbf{0}}^{=}}(\mathcal{D}^{(n-1)}) > \ldots > h_{\mathbf{HB}_{\mathbf{0}}^{=}}(\mathcal{D}^{(1)}) \,,$$

and we will then construct a derivation $\mathcal{D}^{(n+1)}$ in $\mathbf{HB}_{\mathbf{0}}^{=} + \mathrm{SYMM} + \{\mathrm{TRANS}_k\}_k$ such that the extended sequence $\mathcal{SD}^{(n+1)} = \langle \mathcal{D}^{(0)}, \ldots, \mathcal{D}^{(n+1)} \rangle$ fulfills, for $n+1$ instead of n, the conditions (i)′–(vii)′ above with the possible exception of (v)′ (if $\mathcal{D}^{(n+1)}$ is namely a $\mathbf{HB}_{\mathbf{0}}^{=}$-derivation). Then $\mathcal{SD}^{(n+1)}$ can either be taken as the desired sequence $\mathcal{SD}$, if $\mathcal{D}^{(n+1)}$ does not contain SYMM- nor $\{\mathrm{TRANS}_k\}_k$-applications, or otherwise $\mathcal{SD}^{(n+1)}$ again fulfills (i)′–(vii)′, but with $n+1$ instead of n. It is easy to see that the result of this inductive definition is a desired sequence $\mathcal{SD} = \langle \mathcal{D}^{(n)} \rangle_{n \in I}$ of derivations in $\mathbf{HB}_{\mathbf{0}}^{=} + \mathrm{SYMM} + \{\mathrm{TRANS}_k\}_k$, where I is as above mentioned, that fulfills the properties (i)–(vi) in Lemma 8.2.13.

Case 1. $n = 0$ (Base Case).

Here we let $\mathcal{D}^{(0)}$ be the derivation $\mathcal{D}$, and $\mathcal{SD}^{(0)} = \langle \mathcal{D}^{(0)} \rangle$. If $\mathcal{D}^{(0)}$ contains neither SYMM- nor $\{\mathrm{TRANS}_k\}_k$-applications, then we stop with the construction of $\mathcal{SD}$ and let it simply be $\mathcal{SD} = \mathcal{SD}^{(0)}$; $\mathcal{SD}$ obviously satisfies the conditions (i)–(vi) in the lemma (in (v) let $n_{\max} = 0$). Otherwise $\mathcal{SD}^{(0)}$ clearly fulfills conditions (i)′–(vii)′, for $n = 0$.

Case 2. $n = 1$ (Induction Step from $n = 0 \rightarrow n = 1$).

By the induction hypothesis on $\mathcal{SD}^{(n)}$, $\mathcal{D}^{(0)}$ contains SYMM- or $\{\mathrm{TRANS}_k\}_k$-applications, but does not have open assumptions. Therefore Lemma 8.2.18 can be applied: it implies that $\mathcal{D}^{(0)}$ can effectively be transformed, by a proof-theoretic transformation, into a derivation $\mathcal{D}^{(0)\prime}$ in the system $\mathbf{HB}_{\mathbf{0}}^{=} +$ $+ \mathrm{SYMM} + \{\mathrm{TRANS}_k\}_k$ that mimics $\mathcal{D}^{(0)}$ and that is of a form (8.43) such that (i)–(iv) in Lemma 8.2.18 are fulfilled for the occurring designations. By removing from $\mathcal{D}^{(0)\prime}$ all pairs of successive REN-applications, and by eliminating all nlμb-decreasing applications of $\mathrm{FOLD}_{l/r}$ according to the transformation described in the proof of item (iii) of Lemma C.10, Appendix C, we reach effectively a derivation $\mathcal{D}^{(1)}$ in $\mathbf{HB}^{=}_{\mathbf{0}} + \mathrm{SYMM} + \{\mathrm{TRANS}_k\}_k$ that is still of the form (8.43) with (i)–(iv) in Lemma 8.2.18 fulfilled for the occurring designations, but that now satisfies the conditions (I), (III), and (IV) in Lemma 8.2.19 ($\mathcal{D}^{(1)}$ might however violate (II) in this lemma in case that it does not contain SYMM- or $\{\mathrm{TRANS}_k\}_k$-applications any more). Now we let $\mathcal{SD}^{(1)} = \langle \mathcal{D}^{(0)}, \mathcal{D}^{(1)} \rangle$, and find that $\mathcal{SD}^{(1)}$ fulfills, for $n = 1$, the conditions (i)′–(vii)′ with the possible exception of (v)′.

If $\mathcal{D}^{(1)}$ contains neither SYMM- nor $\{\mathrm{TRANS}_k\}_k$-applications, then we stop the inductive definition of $\mathcal{SD}$ and let $\mathcal{SD} = \mathcal{SD}^{(1)}$; $\mathcal{SD}$ then fulfills the conditions (i)–(vi) in the lemma (in (v) let $n_{\max} = 1$). If, however, $\mathcal{D}^{(1)}$ contains SYMM- or $\{\mathrm{TRANS}_k\}_k$-applications, then $\mathcal{D}^{(1)}$ fulfills the conditions (I)–(IV) in Lemma 8.2.19, and as a consequence, $\mathcal{SD}^{(1)}$ fulfills again the properties (i)′–(vii)′, for $n = 1$.

Case 3. $n > 1$ (Induction Step from $n \to n+1$).

By the induction hypothesis we assume a sequence $\mathcal{SD}^{(n)} = \langle \mathcal{D}^{(0)}, \ldots, \mathcal{D}^{(n)} \rangle$ in $\mathbf{HB}^{=}_{\mathbf{0}} + \mathrm{SYMM} + \{\mathrm{TRANS}_k\}_k$ with the properties (i)′–(vii)′ to be given.

Due to the fulfilledness of (ii)′, (iii)′, (v)′, and (vi)′ for $\mathcal{SD}^{(n)}$, it follows that $\mathcal{D}^{(n)}$ satisfies the hypothesis (I)–(IV) of Lemma 8.2.19. Hence this lemma can be applied to $\mathcal{D}^{(n)}$, and thus $\mathcal{D}^{(n)}$ can be transformed effectively, by a proof-theoretic transformation, into a derivation $\mathcal{D}^{(n+1)}$ in $\mathbf{HB}^{=}_{\mathbf{0}} + \mathrm{SYMM} +$ $+ \{\mathrm{TRANS}_k\}_k$ that mimics $\mathcal{D}^{(n)}$, that is of the form (8.43) such that (i)–(iv) in Lemma 8.2.18 are fulfilled for some $m \in \omega \backslash \{0\}$ and $\mathcal{DC} \in \mathcal{D}er\mathcal{C}txt_{\mathrm{m}}(\mathbf{HB}^{=}_{\mathbf{0}})$, and that does not contain successive applications of REN nor nlμb-decreasing applications of $\mathrm{FOLD}_{l/r}$; and furthermore $\mathcal{D}^{(n+1)}$ fulfills

$$h_{\mathbf{HB}^{=}_{\mathbf{0}}}(\mathcal{D}^{(n+1)}) > h_{\mathbf{HB}^{=}_{\mathbf{0}}}(\mathcal{D}^{(n)}) \tag{8.51}$$

if $\mathcal{D}^{(n+1)}$ contains applications of SYMM or $\{\mathrm{TRANS}_k\}_k$ (otherwise the $\mathbf{HB}^{=}_{\mathbf{0}}$-height of $\mathcal{D}^{(n+1)}$ is not defined). We let $\mathcal{SD}^{(n+1)} = \langle \mathcal{D}^{(0)}, \ldots, \mathcal{D}^{(n)}, \mathcal{D}^{(n+1)} \rangle$, and find now that $\mathcal{SD}^{(n+1)}$ fulfills, with $n+1$ in place of n, the conditions (i)′–(vii)′ except possibly (v)′ (since $\mathcal{D}^{(n+1)}$ might already be a $\mathbf{HB}^{=}_{\mathbf{0}}$-derivation): this follows from the induction hypothesis, the construction of $\mathcal{D}^{(n+1)}$ and by (8.51).

If $\mathcal{D}^{(n+1)}$ does not contain applications of SYMM or $\{\mathrm{TRANS}_k\}_k$, then we stop with the construction of the sequence $\mathcal{SD}$ and let $\mathcal{SD} = \mathcal{SD}^{(n+1)}$; as a consequence of the fulfilledness, for $n+1$ instead of n, of conditions (i)′–(vii)′ except (v)′ for $\mathcal{SD}^{(n+1)}$, and of the choice of $\mathcal{D}^{(n+1)}$ and the induction hypothesis on $\mathcal{SD}^{(n)}$, it follows now that the sequence $\mathcal{SD}$ satisfies the conditions (i)–(vi) in the lemma.

If, however, $\mathcal{D}^{(n+1)}$ contains again applications of SYMM or $\{\mathrm{TRANS}_k\}_k$, then the sequence $\mathcal{SD}^{(n+1)}$ also fulfills (v)′, and hence now satisfies the conditions (i)′–(vii)′, with $n+1$ in place of n.

By this inductive definition we have effectively produced a sequence $\mathcal{SD} = \langle \mathcal{D} \rangle_{i \in I}$

of derivations in $\mathbf{HB}_{\mathbf{0}}^{=} + \mathrm{SYMM} + \{\mathrm{TRANS}_k\}_k$ with the properties (i)–(vi) as listed in the lemma.

□

Having proved Lemma 8.2.21 as an auxiliary statement for justifying step (St2)′ in our proof later of Theorem 8.2.2, we continue by stating and proving a lemma that will enable us to perform step (St3)′ in that proof. The assertion of the lemma below is a reformulation with respect to derivations in $\mathbf{HB}_{\mathbf{0}}^{=} + \mathrm{SYMM} + \{\mathrm{TRANS}_k\}_k$, and also a refinement into a precise statement, of step (St2) in the simplified summary given earlier of the transformation developed here between derivations in $\mathbf{HB}^{=}$ and derivations in $\mathbf{HB}_{\mathbf{0}}^{=}$.

Lemma 8.2.21. *Let $\tau, \sigma \in \mu Tp$. Furthermore, let $\mathcal{SD} = \langle \mathcal{D}^{(n)} \rangle_{n\in\omega}$ be an infinite sequence of derivations in $\mathbf{HB}_{\mathbf{0}}^{=} + \mathrm{SYMM} + \{\mathrm{TRANS}_k\}_k$ that has the following properties:*

- *all derivations $\mathcal{D}^{(n)}$ in $\mathcal{SD}$ have conclusion $\tau = \sigma$ and do not possess open assumptions,*
- *all derivations $\mathcal{D}^{(n)}$ in $\mathcal{SD}$ contain* SYMM- *or* $\{\mathrm{TRANS}_k\}_k$*-applications,*
- *for all $n \in \omega$, $\mathcal{D}^{(n)}$ does not contain successive applications of* REN, *nor nlμb-decreasing applications of* FOLD_k *or* FOLD_r*,*
- *for all $n \in \omega$, the derivation $\mathcal{D}^{(n+1)}$ is the result of applying an effective proof-theoretic transformation to $\mathcal{D}^{(n)}$,*
- *the $\mathbf{HB}_{\mathbf{0}}^{=}$-heights $h_{\mathbf{HB}_{\mathbf{0}}^{=}}(\mathcal{D}^{(n)})$ of the derivations $\mathcal{D}^{(n)}$ tend to infinity, that is, put symbolically, $\left\langle h_{\mathbf{HB}_{\mathbf{0}}^{=}}(\mathcal{D}^{(n)}) \right\rangle_{n\in\omega} \to_{n\to\infty} \infty$ holds.*

Then for all $n \in \omega$ with the property $h_{\mathbf{HB}_{\mathbf{0}}^{=}}(\mathcal{D}^{(n)}) > h_{(\tau,\sigma)}$, where

$$h_{(\tau,\sigma)} =_{\mathrm{def}} 2\left((s(\tau)+1)(s(\sigma)+1) + 2|\tau| + 2|\sigma| + 2\right) , \tag{8.52}$$

a $\mathbf{HB}_{\mathbf{0}}^{=}$-end-derivation $\lceil \mathcal{D}^{(n)} \rceil_{\mathbf{HB}_{\mathbf{0}}^{=}}^{\{u_i\}_i}$ of $\mathcal{D}^{(n)}$, where $\{u_i\}_i$ is an appropriate family of distinct assumption markers not occurring in $\mathcal{D}^{(n)}$, can effectively be transformed, by the transformation described in the proof of Theorem C.14 in Appendix C, into a derivation $\mathcal{D}^{(n)\prime}$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ without open assumptions and with conclusion $\tau = \sigma$.

And in particular, a derivation $\mathcal{D}'$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ with conclusion $\tau = \sigma$, without open assumptions, and with depth $|\mathcal{D}'| < h_{(\tau,\sigma)}$ can effectively be produced by a proof-theoretic transformation from $\mathcal{D}^{(0)}$, the first derivation in the sequence $\mathcal{SD}$.

Proof. Let $\tau, \sigma \in \mu Tp$ be arbitrary and let $\mathcal{SD} = \langle \mathcal{D}^{(n)} \rangle_{n\in\omega}$ be a sequence of derivations in $\mathbf{HB}_{\mathbf{0}}^{=} + \mathrm{SYMM} + \{\mathrm{TRANS}_k\}_k$ as in the hypothesis of the lemma. And let $h_{(\tau,\sigma)}$ be defined as in (8.52).

By assumption on $\mathcal{SD}$ $\left\langle h_{\mathbf{HB}^{=}_{\mathbf{0}}}(\mathcal{D}^{(n)})\right\rangle_{n\in\omega} \to_n \infty$ is the case, i.e. the minimal heights $h_{\mathbf{HB}^{=}_{\mathbf{0}}}(\mathcal{D}^{(n)})$ in $\mathcal{D}^{(n)}$ of conclusions of rules SYMM or $\{\text{TRANS}_k\}_k$ tend to infinity. Due to this there exist infinitely many $n \in \omega$ with the property

$$h_n \;\geq\; h_{(\tau,\sigma)}\;. \tag{8.53}$$

Let now $n \in \omega$ be arbitrary such that (8.53) holds. Furthermore, let $m \in \omega\backslash\{0\}$ be such that $\lceil\mathcal{D}^{(n)}\rceil^{\{[\,]_i\}_i}_{\mathbf{HB}^{=}_{\mathbf{0}}} \in \mathcal{D}\mathit{er}\mathcal{C}\mathit{txt}_m(\mathbf{HB}^{=}_{\mathbf{0}})$, and let $\{u_i\}_{i=1,\ldots,m}$ be a family of distinct assumption markers that do not occur in $\mathcal{D}^{(n)}$. Then the hypotheses of Corollary C.15 in Appendix C.11 are satisfied for a $\mathbf{HB}^{=}_{\mathbf{0}}$-end-derivation $\lceil\mathcal{D}^{(n)}\rceil^{\{u_i\}_i}_{\mathbf{HB}^{=}_{\mathbf{0}}}$ of $\mathcal{D}^{(n)}$: conditions (i) and (ii) in Theorem C.11 are satisfied for $\lceil\mathcal{D}^{(n)}\rceil^{\{u_i\}_i}_{\mathbf{HB}^{=}_{\mathbf{0}}}$ because they are satisfied for $\mathcal{D}^{(n)}$ by assumption on $\mathcal{SD}$; and furthermore, due to (8.53) and the fact that $\mathcal{D}^{(n)}$ is a derivation in $\mathbf{HB}^{=}_{\mathbf{0}}+\text{SYMM}+\text{TRANS}$ without open assumptions, it follows that $\lceil\mathcal{D}^{(n)}\rceil^{\{u_i\}_i}_{\mathbf{HB}^{=}_{\mathbf{0}}}$ contains open assumptions only at heights $\geq h_{(\tau,\sigma)}$. Therefore Corollary C.14 can be applied; it entails that $\lceil\mathcal{D}^{(n)}\rceil^{\{u_i\}_i}_{\mathbf{HB}^{=}_{\mathbf{0}}}$ can effectively be transformed into a derivation $\mathcal{D}^{(n)\prime}$ in $\mathbf{HB}^{=}_{\mathbf{0}}$ without open assumptions, with conclusion $\tau=\sigma$ and with depth $|\mathcal{D}^{(n)\prime}| < h_{(\tau,\sigma)}$.

Since there exists an $n \in \omega$ such that (8.53) holds (there are in fact infinitely many such n), it is possible, by successively generating derivations $\mathcal{D}^{(j)}$ of the sequence $\mathcal{DC}$ and by checking whether or not $h_j \geq h_{(\tau,\sigma)}$ holds, to find effectively a first $n_0 \in \omega$ with the property $h_{n_0} \geq h_{(\tau,\sigma)}$. By what we have already shown above, it follows that $\lceil\mathcal{D}^{(n_0)}\rceil^{\{u_i\}_i}_{\mathbf{HB}^{=}_{\mathbf{0}}}$ can effectively be transformed into a derivation $\mathcal{D}'$ in $\mathbf{HB}^{=}_{\mathbf{0}}$ without open assumptions, with conclusion $\tau=\sigma$ and with depth $|\mathcal{D}'| < h_{(\tau,\sigma)}$. Hence a derivation $\mathcal{D}'$ with these properties can effectively be generated from the presupposed sequence $\mathcal{SD}$ of derivations, and hence from the first derivation $\mathcal{D}^{(0)}$ in $\mathcal{SD}$ because by assumption on $\mathcal{SD}$ each derivation in this sequence can effectively be produced from its predecessor derivation.

□

We are now finally able to give the proof for Theorem 8.2.2, the main theorem in this section about a proof-theoretic transformation between derivations in the systems $\mathbf{HB}^{=}$ and $\mathbf{HB}^{=}_{\mathbf{0}}$.

Proof of Theorem 8.2.2. Let $\mathcal{D}$ be a derivation in $\mathbf{HB}^{=}$ without open assumptions and with conclusion $\tau=\sigma$, for some $\tau,\sigma \in \mu Tp$. We have to show that $\mathcal{D}$ can be transformed, by an effective proof-theoretic procedure, into a derivation $\mathcal{D}'$ in $\mathbf{HB}^{=}_{\mathbf{0}}$ without open assumptions and with the same conclusion $\tau=\sigma$ as $\mathcal{D}$.

(St1)′ Applying Lemma 8.2.3, $\mathcal{D}$ can effectively be transformed, in a proof-theoretic way, into a derivation $\tilde{\mathcal{D}}$ in $\mathbf{HB}^{=}_{\mathbf{0}}+\text{SYMM}+\text{TRANS}$ without open assumptions and with the same conclusion $\tau=\sigma$ as $\mathcal{D}$. Let $\tilde{\mathcal{D}}^{(0)}$ be a derivation that is found in this way. By furthermore renaming all applications of the transitivity rule TRANS in $\tilde{\mathcal{D}}^{(0)}$ into applications of the generalized transitivity rule TRANS_1, we reach effectively a mimicking derivation $\tilde{\mathcal{D}}$ for $\mathcal{D}$ in

$\mathbf{HB}_{\mathbf{0}}^{=} + \text{SYMM} + \{\text{TRANS}_k\}_k$ without open assumptions and with conclusion $\tau = \sigma$.

(St2)′ Due to Lemma 8.2.13, there exists a sequence $\mathcal{SD} = \langle \mathcal{D}^{(n)} \rangle_{n \in I}$, where $I = \omega$ or $I = [0, n_{\max}] \cap \omega$ for some $n_{\max} \in \omega$, of derivations in $\mathbf{HB}_{\mathbf{0}}^{=} + \text{SYMM} + \{\text{TRANS}_k\}_k$ that starts with $\mathcal{D}^{(0)} = \tilde{\mathcal{D}}$ and that fulfills the conditions (ii)–(vi) in Lemma 8.2.13. Let $\mathcal{SD}$ be such a sequence. Because of condition (iv) in Lemma 8.2.13, it holds in particular that, for all $n \in I$ with $n+1 \in I$, the derivation $\mathcal{D}^{(n+1)}$ is the result of applying a proof-theoretical procedure to the derivation $\mathcal{D}^{(n)}$. Now we distinguish two cases depending on whether $\mathcal{SD}$ is finite or infinite.

If the sequence $\mathcal{SD}$ is finite, i.e. if $I = [0, n_{\max}] \cap \omega$ for some $n_{\max} \in \omega$, then it ends (since condition (v) in Lemma 8.2.13 is fulfilled for $\mathcal{SD}$) with a derivation $\mathcal{D}^{(n_{\max})}$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ without open assumptions and with the conclusion $\tau = \sigma$, the same as that of $\tilde{\mathcal{D}}$ and of $\mathcal{D}$. By overlooking and gathering all effective steps carried out during the sequence $\mathcal{SD}$ and the initial transformation step from $\mathcal{D}$ to $\tilde{\mathcal{D}}$, we can say that $\mathcal{D}^{(n_{\max})}$ is the result of applying an effective proof-theoretic transformation to $\mathcal{D}$. In this case we can take $\mathcal{D}^{(n_{\max})}$ as the desired mimicking derivation $\mathcal{D}'$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ for $\mathcal{D}$.

The case in which $\mathcal{SD}$ is infinite is treated in (St3).

(St3)′ If the sequence $\mathcal{SD}$ is infinite, then it fulfills, due to its choice, the hypotheses of Lemma 8.2.21. Hence this lemma can be invoked to show that a derivation $\mathcal{D}'$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ without open assumptions and with the same conclusion as $\mathcal{D}$ can effectively be produced from $\tilde{\mathcal{D}}$ by a proof-theoretic transformation. Let $\mathcal{D}'$ be a derivation that is found in this way. Because the initial step from $\mathcal{D}$ to $\tilde{\mathcal{D}}$ has been of an effective proof-theoretic character, too, it follows that $\mathcal{D}'$ can also be produced from $\mathcal{D}$ by an effective proof-theoretic transformation.

In both cases that are able to occur in (St2)′, of $\mathcal{SD}$ being either finite or infinite, we have effectively produced from $\mathcal{D}$, by a proof-theoretic transformation, a derivation $\mathcal{D}'$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ without open assumptions and with the same conclusion $\tau = \sigma$ as $\mathcal{D}$. □

We close this section with the following observation: it is easy to see that the transformation we have demonstrated to exist in the proof of Theorem 8.2.2 can be taken to pieces and then reassembled more elegantly again with the algorithm defined by the flow-chart in Figure 8.4 as the result.

Chapter 9

Conclusion

In this final chapter we have two goals: firstly, to summarize the transformations between proof systems developed in previous chapters, as well as to discuss some specific issues concerning the way how these transformations interact and are related, and secondly, to outline directions for possible extensions of the results we have obtained. These issues are treated separately in Section 9.1 and in Section 9.2.

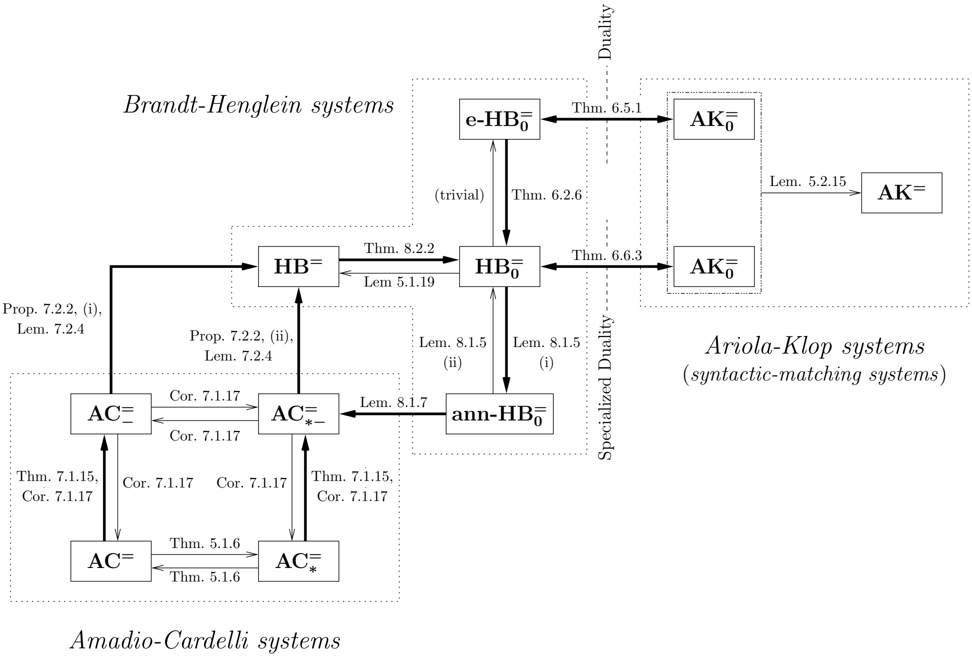
9.1 The Obtained Network of Transformations

In Figure 9.1 a schematic overview is given about how the proof systems of the three kinds of systems introduced in Chapter 5 and in later chapters are linked by the proof-theoretic transformations that we have developed. The Amadio-Cardelli systems $\mathbf{AC}^{=}$, $\mathbf{AC}^{=}_{*}$, $\mathbf{AC}^{=}_{-}$, and $\mathbf{AC}^{=}_{*-}$, the Brandt-Henglein systems $\mathbf{HB}^{=}$, $\mathbf{HB}^{=}_{\mathbf{0}}$, $\mathbf{e}$-$\mathbf{HB}^{=}_{\mathbf{0}}$, and $\mathbf{ann}$-$\mathbf{HB}^{=}_{\mathbf{0}}$, and the syntactic-matching systems (Ariola-Klop systems) $\mathbf{AK}^{=}$ and $\mathbf{AK}^{=}_{\mathbf{0}}$ have been arranged into a 'network' whose 'wirings' represent transformations described in the Chapters 5–8. Hereby references to respective existence statements for the transformations have been attached to the arrows representing them, accordingly.

Thin arrows have been used to symbolize easy transformations, and thick arrows for more involved ones. For example, the transformation between derivations in $\mathbf{AC}^{=}$ into mimicking derivations in $\mathbf{AC}^{=}_{*}$ is considered "easy" because it proceeds by an elimination procedure of UFP-applications which is comparatively simple owing to the fact that UFP is a derivable rule of $\mathbf{AC}^{=}_{*}$; in contrast with this, for instance the transformation from derivations in $\mathbf{AC}^{=}$ into mimicking derivations in $\mathbf{AC}^{=}_{-}$ that proceeds by μ-COMPAT-elimination as described in the proof of Theorem 7.1.15 is certainly "more involved", which also reflects the fact that μ-COMPAT is merely admissible, but not derivable, in $\mathbf{AC}^{=}_{-}$.

It has to be mentioned that the arrows in both ways between $\mathbf{e}$-$\mathbf{HB}^{=}_{\mathbf{0}}$ and $\mathbf{AK}^{=}_{\mathbf{0}}$, and between $\mathbf{HB}^{=}_{\mathbf{0}}$ and $\mathbf{AK}^{=}_{\mathbf{0}}$ represent different kind of transformations than the other arrows. They symbolize transformations that 'implement' the reflection

Figure 9.1: ‘Connecting’ proof systems for recursive type equality: the ‘network’ of the developed transformations between derivations in the Amadio-Cardelli systems $\mathbf{AC}^{=}$, $\mathbf{AC}^{=}_{*}$, $\mathbf{AC}^{=}_{-}$, and $\mathbf{AC}^{=}_{*-}$, derivations in the Brandt-Henglein systems $\mathbf{HB}^{=}$, $\mathbf{HB}^{=}_{\mathbf{0}}$, $\mathbf{e\text{-}HB}^{=}_{\mathbf{0}}$, and $\mathbf{ann\text{-}HB}^{=}_{\mathbf{0}}$, and consistency-unfoldings, and respectively derivations, in the Ariola-Klop systems $\mathbf{AK}^{=}_{\mathbf{0}}$ and $\mathbf{AK}^{=}$.



functions $\mathcal{C}$ and $\mathcal{D}$ between derivations in $\mathbf{e\text{-}HB}^{=}_{\mathbf{0}}$ and consistency-unfoldings in $\mathbf{AK}^{=}_{\mathbf{0}}$, and respectively, between derivations in $\mathbf{HB}^{=}_{\mathbf{0}}$ and consistency-unfoldings in $\mathbf{AK}^{=}_{\mathbf{0}}$ (with the property **D**). In contrast with this, each of the other arrows in Figure 9.1 denotes a transformation between derivations in the proof system at the arrow’s source into derivations in the system at its target.

An interesting observation is related to the position of the variant proof systems $\mathbf{AC}^{=}_{-}$ and $\mathbf{AC}^{=}_{*-}$ of $\mathbf{AC}^{=}$ and $\mathbf{AC}^{=}_{*}$ within the ‘network’ in Figure 9.1 of the transformations that we have given; it concerns the way in which the transformations from $\mathbf{AC}^{=}$ to $\mathbf{HB}^{=}$ and, vice versa, from $\mathbf{HB}^{=}$ to $\mathbf{AC}^{=}$ have actually been found. On the one hand, each of the transformations, corresponding to paths in the ‘network’ in Figure 9.1, from derivations in $\mathbf{AC}^{=}$ into derivations in $\mathbf{HB}^{=}$ proceeds via derivations in one of the μ-COMPAT-free variant systems $\mathbf{AC}^{=}_{-}$ and $\mathbf{AC}^{=}_{*-}$ of $\mathbf{AC}^{=}$ and $\mathbf{AC}^{=}_{-}$. And on the other hand, also all transformations, made possible by the results given here, between derivations in $\mathbf{HB}^{=}$ and derivations in $\mathbf{AC}^{=}$ proceed via intermediary results of derivations in $\mathbf{AC}^{=}_{-}$ and $\mathbf{AC}^{=}_{*-}$.[1] The first fact mentioned is merely a consequence of the way how we have constructed the transformation from $\mathbf{AC}^{=}$ to $\mathbf{HB}^{=}$: by setting out to eliminate μ-COMPAT-applications from $\mathbf{AC}^{=}$-derivations always right away, which choice of procedure has lead us to the introduction of the systems $\mathbf{AC}^{=}_{-}$ and $\mathbf{AC}^{=}_{*-}$ in the first place. Contrasting with this, the second fact came as a slight surprise, later, after developing the transformation from $\mathbf{HB}^{=}$-derivations via **ann-**$\mathbf{HB}^{=}_{\mathbf{0}}$-derivations into $\mathbf{AC}^{=}$-derivations described in Chapter 8, Section 8.1: only then did we notice that the $\mathbf{AC}^{=}$-derivations this transformation produces do not contain applications of μ-COMPAT, or put as a stronger assertion, that this transformation produces derivations in $\mathbf{AC}^{=}_{*-}$ *in a natural way*.[2] This observation lends some legitimacy to the claim we want to make here that the μ-COMPAT-free variant systems $\mathbf{AC}^{=}_{*-}$ and $\mathbf{AC}^{=}_{-}$ can be viewed as axiomatizations of recursive type equality that may be viewed to be more 'compact' than $\mathbf{AC}^{=}$ and $\mathbf{AC}^{=}_{*}$ (because $\mathbf{AC}^{=}_{*-}$ and $\mathbf{AC}^{=}_{-}$ contain fewer rules), but that are equally 'natural'.

The network of transformations in Figure 9.1 also reflects the facts that the transformation from $\mathbf{HB}^{=}$-derivations into $\mathbf{AC}^{=}$-derivations developed in Chapter 8 makes a 'detour' via derivations in the analytic variant system $\mathbf{HB}^{=}_{\mathbf{0}}$ of $\mathbf{HB}^{=}$ and derivations in the annotated version **ann-**$\mathbf{HB}^{=}_{\mathbf{0}}$ of $\mathbf{HB}^{=}_{\mathbf{0}}$, and that we have not developed a more direct transformation: there is no arrow pointing directly from $\mathbf{HB}^{=}$ to $\mathbf{AC}^{=}$. Therefore the only possibility for an effective transformation from $\mathbf{HB}^{=}$-derivations into $\mathbf{AC}^{=}$-derivations enabled by the results given here is to 'normalize' a given $\mathbf{HB}^{=}$-derivation in the first step by applying the procedure detailed in Section 8.2, which is rather involved. In Remark 8.1.10, at the end of Section 8.1, it has already been mentioned that we do not know presently how to construct a more direct transformation from $\mathbf{HB}^{=}$ to $\mathbf{AC}^{=}$, one that avoids the complicated process of 'normalizing' $\mathbf{HB}^{=}_{\mathbf{0}}$-derivations. We pose this as an Open Problem here.

Open Problem 9.1.1. Does there exist *a more direct* effective transformation from derivations $\mathcal{D}$ in $\mathbf{HB}^{=}$ without open assumptions into derivations $\mathcal{D}'$ in $\mathbf{AC}^{=}$ without assumptions and with the same conclusion as $\mathcal{D}$ *than* the one that is justified by the results in Chapter 8 (and that proceeds via derivations in the analytic variant $\mathbf{HB}^{=}_{\mathbf{0}}$ of $\mathbf{HB}^{=}$ and the annotated version **ann-**$\mathbf{HB}^{=}_{\mathbf{0}}$ of $\mathbf{HB}^{=}_{\mathbf{0}}$)? Furthermore, can the approach of first assigning appropriate annotations, and of subsequently extracting $\mathbf{AC}^{=}$-derivations be generalized from $\mathbf{HB}^{=}_{\mathbf{0}}$-derivations to $\mathbf{HB}^{=}$-deriva-

[1] The arrows in the 'network' even tell us that all such transformations proceed via intermediary results of derivations in $\mathbf{AC}^{=}_{*-}$. This is indeed the case for the transformations justified by our proofs. However, these proofs could easily be altered slightly to produce $\mathbf{AC}^{=}_{-}$-derivations instead of $\mathbf{AC}^{=}_{*-}$-derivations and in this manner to circumvent $\mathbf{AC}^{=}_{*-}$-derivations. In particular, Lemma 8.1.6 could be proved directly and analogously to Lemma 8.1.7, in place of being shown as an immediate consequence of Lemma 8.1.7, thereby creating a detour via $\mathbf{AC}^{=}_{*-}$-derivations.

[2] Also, it takes only little effort to modify this transformation into producing $\mathbf{AC}^{=}_{-}$-derivations without μ-COMPAT-applications, or into producing $\mathbf{AC}^{=}_{*}$- or $\mathbf{AC}^{=}$-derivations (of the specific form that they contain μ-COMPAT applications only above such formulas that occur in the axiom scheme $(\mu\!-\!\bot)'$).

tions *in some way*?

In Remark 8.1.10, we have also mentioned that the transformation of $\mathbf{HB}_{\mathbf{0}}^{=}$-derivations into $\mathbf{AC}^{=}$-derivations explained in Section 8.1 can be extended to an analogous transformation from derivations in the extension $\mathbf{e\text{-}HB}_{\mathbf{0}}^{=}$ of $\mathbf{HB}_{\mathbf{0}}^{=}$ into derivations in $\mathbf{AC}^{=}$, in such a way that the more general transformation proceeds via derivations in an annotated version $\mathbf{ann\text{-}e\text{-}HB}_{\mathbf{0}}^{=}$ of $\mathbf{e\text{-}HB}_{\mathbf{0}}^{=}$ and derivations in the μ-COMPAT-free variant system $\mathbf{AC}_{*-}^{=}$ of $\mathbf{AC}_{*}^{=}$ and $\mathbf{AC}^{=}$. Since this result has not been proved here (also the annotated version $\mathbf{ann\text{-}e\text{-}HB}_{\mathbf{0}}^{=}$ of $\mathbf{e\text{-}HB}_{\mathbf{0}}^{=}$ has not been formally introduced), corresponding arrows have not been taken up into the network of transformations in Figure 9.1.

9.2 Directions for Possible Extensions

In this section we describe a number of directions into which the work presented here about proof-theoretic transformations between recursive type equality can be, or is likely to be, generalized. Hereby we are also going to report informally about results that we have obtained, but that have not been published already.

9.2.1 Brandt-Henglein Systems with More Circular Rules

In Chapter 6, Section 6.2, we introduced an extension $\mathbf{e\text{-}HB}_{\mathbf{0}}^{=}$ of the analytic Brandt-Henglein system $\mathbf{HB}_{\mathbf{0}}^{=}$ by adding new 'circular' rules, a system in which there are also other rules that formalize coinductive reasoning apart from the familiar rule ARROW/FIX of the axiomatization of Brandt and Henglein for recursive type equality. For this purpose we first introduced a general such rule FIX that allows applications of the form

$$\begin{array}{c} [\tau = \sigma]^{u} \\ \mathcal{D}_1 \\ \dfrac{\tau = \sigma}{\tau = \sigma}\ \text{FIX},\ u \quad \text{(if side-condition (9.2))} \end{array} \tag{9.1}$$

with the indicated *proviso* that the immediate subderivation $\mathcal{D}_1$ is *contractive* with respect to the open marked assumptions that get discharged by this FIX-application, i.e. open marked assumptions in $\mathcal{D}_1$ of the form $(\tau = \sigma)^{u}$; by this, the restriction

> "For each open marked assumption of the form $(\tau = \sigma)^{u}$ in $\mathcal{D}_1$, the thread down to the conclusion of $\mathcal{D}_1$ *crosses* an application of ARROW or ARROW/FIX *at least once*." (9.2)

was meant. Then we showed that FIX is cr-admissible in $\mathbf{HB}_{\mathbf{0}}^{=}$ as well as that applications of FIX can be eliminated effectively from every derivation $\mathcal{D}$ in $\mathbf{HB}_{\mathbf{0}}^{=}$+FIX with the result of a mimicking derivation for $\mathcal{D}$ in $\mathbf{HB}_{\mathbf{0}}^{=}$. And subsequently, we defined 'circular' variants of the rules REN, FOLD$_l$ and FOLD$_r$ in $\mathbf{HB}_{\mathbf{0}}^{=}$, the rules REN/FIX, FOLD$_l$/FIX, and FOLD$_r$/FIX, and used cr-admissibility of FIX in $\mathbf{HB}_{\mathbf{0}}^{=}$

to show cr-admissibility in $\mathbf{HB}_{\mathbf{0}}^{=}$ also for these new rules. Eventually, we defined an extension $\mathbf{e\text{-}HB}_{\mathbf{0}}^{=}$ of $\mathbf{HB}_{\mathbf{0}}^{=}$ by adding these three new rules, and proved that $\mathbf{e\text{-}HB}_{\mathbf{0}}^{=}$ and $\mathbf{HB}_{\mathbf{0}}^{=}$ are equivalent (and hence that also $\mathbf{e\text{-}HB}_{\mathbf{0}}^{=}$ is a sound and complete axiomatization for $=_\mu$).

We have recalled these results here for the following reason: the question can be asked whether the basic Brandt-Henglein system $\mathbf{HB}^{=}$ can be extended in a similar manner by adding rules of an analogous kind (such that, in particular, no new theorems become derivable); of if there did actually exist a substantial reason for why in the axiomatization of Brandt and Henglein only one ‘circular’ rule figures, namely the rule ARROW/FIX.

The answers that we want to give here to the two parts of the question just formulated are “yes” and “no”, respectively. Below we report, without giving proofs, on results that we have obtained concerning these issues.

- It turns out that the rule FIX with applications of the form (9.1), where (9.2) is fulfilled, is actually not only cr-admissible in $\mathbf{HB}_{\mathbf{0}}^{=}$, but also in $\mathbf{HB}^{=}$. This can be shown by a similar, albeit rather more complicated, proof than the one we gave for Lemma 6.2.3: derivations in $\mathbf{HB}^{=}$+FIX can be ‘unfolded’ above a considered application of FIX in a certain manner, using a more-step process and utilizing the deductive power of the rule ARROW/FIX in $\mathbf{HB}^{=}$ to discharge newly arising open assumptions at specific positions.

- Relying on cr-admissibility of FIX in $\mathbf{HB}^{=}$, other ‘circular’ rules can be shown to be cr-admissible in $\mathbf{HB}^{=}$ as well: this holds, in particular, for a variant SYMM/FIX of the rule SYMM of $\mathbf{HB}^{=}$, and for the rule TRANS/FIX with applications of the form

$$\dfrac{\begin{array}{c}[\tau = \sigma]^u\\ \mathcal{D}_1\\ \tau = \rho\end{array} \qquad\qquad \begin{array}{c}[\tau = \sigma]^u\\ \mathcal{D}_2\\ \rho = \sigma\end{array}}{\tau = \sigma}\ \text{TRANS/FIX}, u \qquad (\text{if side-condition } \mathbf{C})\,,$$

  where, analogously as for the rules REN/FIX and $\text{FOLD}_{l/r}$/FIX defined in Definition 6.2.4, the side-condition $\mathbf{C}$ demands that both of the immediate subderivations $\mathcal{D}_1$ and $\mathcal{D}_2$ are contractive with respect to open marked assumptions of the form $(\tau = \sigma)^u$.

- As a consequence, an extension $\mathbf{e\text{-}HB}^{=}$ of $\mathbf{HB}^{=}$ by adding the rules SYMM/FIX and TRANS/FIX can be shown to be equivalent to $\mathbf{HB}^{=}$. Eventually also a system $\mathbf{ext\text{-}HB}^{=}$ that is defined as the union of $\mathbf{e\text{-}HB}_{\mathbf{0}}^{=}$ and $\mathbf{e\text{-}HB}^{=}$ can be proved to be equivalent with the basic Brandt-Henglein system $\mathbf{HB}^{=}$.

And what is more, a close connection can be established between these extensions of the basic Brandt-Henglein system $\mathbf{HB}^{=}$ for recursive type equality and a system for “coercion typing rules” for the subtyping relation $\leq_\mu$ on recursive types that is given by Brandt and Henglein in [BrHe98, Fig.6, p.19]; the latter system also contains a general ‘circular’ FIX-rule.

9.2.2 Syntactic-Matching Tableaux

The definition in Chapter 6 of consistency-unfoldings in a syntactic-matching system as downwards-growing derivation trees with special properties (of containing back-bound leaf-occurrences of marked formulas that fulfill certain conditions) was devised for the special purpose at hand here: for formulating and proving duality statements between consistency-unfoldings in $\mathbf{AK}_{\mathbf{0}}^{=}$ and derivations in $\mathbf{HB}_{\mathbf{0}}^{=}$ or in $\mathbf{e\text{-}HB}_{\mathbf{0}}^{=}$. However, it is not difficult to see that the concept of consistency-unfolding bears a striking analogy with the concept 'closed analytic tableau' as introduced by Smullyan in [Sm68].

Closer inspection of this analogy shows that the two duality theorems stated and proved in Chapter 6, Theorem 6.5.1 and Theorem 6.6.3, lend themselves for being reformulated with respect to an in each case suitably defined tableau calculus. It is possible to prove analogous theorems about a close functional relationship between derivations in $\mathbf{HB}_{\mathbf{0}}^{=}$, or derivations in $\mathbf{e\text{-}HB}_{\mathbf{0}}^{=}$, and so called 'syntactic-matching tableaux' in a respective tableau system. For tentative formulations of results in this direction we refer to [Gra02a], the slides for a talk given at the CWI Amsterdam in May 2002.

9.2.3 Proof Systems for Subtyping on Recursive Types

The three kinds of proof systems for recursive type equality which have been defined in Chapter 5 possess closely related counterparts in proof systems for the subtyping relation $\leq_\mu$ on recursive types that was introduced[3] by Amadio and Cardelli in [AmCa93]. These authors have also given a complete axiomatization for $\leq_\mu$ in what can actually be formulated as a natural-deduction system $\mathbf{AC}^{\leq}$ (see (9.3) below for the rule μ-INTRO of this system, applications of which typically discharge open assumptions); the system $\mathbf{AC}^{\leq}$ contains the axiomatization $\mathbf{AC}^{=}$ introduced in Chapter 5 as its part. A more recent (complete) axiomatization of $\leq_\mu$ has been given by Brandt and Henglein in [BrHe98]; a natural-deduction system formulation $\mathbf{HB}^{\leq}$ of their (sequent-style) system is very similar to the system $\mathbf{HB}^{=}$ introduced in Chapter 5. For this system, an 'analytic' variant system $\mathbf{HB}_{\mathbf{0}}^{\leq}$ can be defined in analogy with the variant system $\mathbf{HB}_{\mathbf{0}}^{=}$ of $\mathbf{HB}^{=}$ introduced in Chapter 5. And finally, also the syntactic-matching proof systems $\mathbf{AK}^{=}$ and $\mathbf{AK}_{\mathbf{0}}^{=}$ possess counterparts in respective syntactic matching systems $\mathbf{AK}^{\leq}$ and $\mathbf{AK}_{\mathbf{0}}^{\leq}$ for $\leq_\mu$: these systems are sound and complete with respect to $\leq_\mu$ in the sense that, for arbitrary $\tau, \sigma \in \mu Tp$, the subtype inequality $\tau \leq \sigma$ can be added consistently[4] to the system if and only if $\tau \leq_\mu \sigma$ holds.

It turns out that most of the transformations that we have developed here between proof systems for recursive type equality $=_\mu$ can be adapted to yield similar

[3] In the words of [BrHe98], "[AmCa93] extend the standard contravariant structural subtyping relation on μ-free types (to be thought of as finite trees) [. . .] in a natural fashion to infinite trees".

[4] For this statement a clarification is needed of what a 'contradiction with respect to $\leq_\mu$' is. It turns out that a definition of this notion is quite directly suggested by the definition of the subtyping relation $\leq_\mu$ due to Amadio and Cardelli.

transformations between the mentioned proof systems for the subtyping relation $\leq_\mu$. Below we report of the following: firstly, of three results that we have obtained concerning adaptations of transformations detailed here to proof systems for $\leq_\mu$ (see items (i)–(iii)); then of a result about the admissibility in $\mathbf{HB}^{\leq}$ of a rule $\mathrm{FIX}^{\leq}$ that is similar to a result only sketched here in Subsection 9.2.1 (confer item (iv)); and lastly, and of an observation which can be paraphrased by saying that there exists a substantially closer proof-theoretic relationship between the Amadio-Cardelli systems and the Brandt-Henglein systems for the subtyping relation $\leq_\mu$ than between the Amadio-Cardelli and Brandt-Henglein systems for recursive type equality $=_\mu$ (see the assertions in item (v) below).

(i) There exists an immediate 'duality' between consistency-unfoldings in $\mathbf{AK}_{\mathbf{0}}^{\leq}$ and derivations without open assumptions in $\mathbf{HB}_{\mathbf{0}}^{\leq}$. This duality has the form of statements similar to Theorem 6.5.1 and Theorem 6.6.3.

(ii) Every derivation $\mathcal{D}$ in $\mathbf{AC}^{\leq}$ without assumptions can be transformed, by an effective proof-theoretic transformation, into a derivation $\mathcal{D}'$ in $\mathbf{HB}^{\leq}$ with the same conclusion as $\mathcal{D}$ and without open assumptions.

(iii) Reversely, the following holds: every derivation $\mathcal{D}$ in $\mathbf{HB}^{\leq}$ without open assumptions can be transformed, by an effective proof-theoretic transformation, into a derivation $\mathcal{D}'$ in $\mathbf{AC}^{\leq}$ with the same conclusion as $\mathcal{D}$ and without assumptions.

(iv) A rule $\mathrm{FIX}^{\leq}$ for a system with inequations between recursive types as formulas that is defined similar to the rule FIX as introduced in Definition 6.2.1 is cr-admissible in $\mathbf{HB}^{\leq}$. What is more, applications of this rule $\mathrm{FIX}^{\leq}$ can effectively be eliminated from derivations in $\mathbf{HB}^{\leq}+\mathrm{FIX}^{\leq}$ with the results of respective mimicking derivations in $\mathbf{HB}^{\leq}$.

(v) There is a close proof-theoretic connection between a particular rule of $\mathbf{AC}^{\leq}$, the rule μ-INTRO, that has applications of the form

$$\frac{\begin{array}{c}[\alpha \leq \beta]^u \\ \mathcal{D}_1 \\ \tau \leq \sigma\end{array}}{\mu\alpha.\,\tau \leq \mu\beta.\,\sigma}\ \mu\text{-INTRO},\, u \quad \text{(if (9.4) is fulfilled)}, \tag{9.3}$$

where the side-condition

$$\left.\begin{array}{l}\text{“}\alpha \text{ occurs only in } \tau\text{, not in } \sigma\,;\ \beta \text{ occurs only in } \sigma\text{, not in } \tau\,;\\ \text{and neither of } \alpha \text{ and } \beta \text{ occurs in an open marked assumption}\\ \text{of } \mathcal{D}_1 \text{ that is different from } (\alpha \leq \beta)^u\,.\text{”}\end{array}\right\} \tag{9.4}$$

has to be observed, with the rule ARROW/FIX in $\mathbf{HB}^{\leq}$ as well as with the rule $\mathrm{FIX}^{\leq}$ mentioned in (iv). To be more precise: μ-INTRO is cr-admissible in $\mathbf{HB}^{\leq}$, and applications of this rule can effectively be eliminated from derivations in $\mathbf{HB}^{\leq}+\mu$-INTRO with the result of respective mimicking derivations.

And reversely, the rules ARROW/FIX and $\text{FIX}^{\leq}$ are cr-admissible in $\mathbf{AC}^{\leq}$ and can be eliminated effectively from derivations in the extension of $\mathbf{AC}^{\leq}$ by adding these rules such that respective mimicking derivations are produced.

Some more information about the transformations and statements asserted in (i)–(v) above can be found on [Gra01], the slides for a talk given at the Catholic University of Nijmegen (KUN) in January 2001.

9.2.4 Proof Systems for Equivalence of Regular Expressions

As an example of quite different, but (respectively) conceptually related proof systems to which we expect that the transformations developed here can be adapted, we mention formal systems for the equivalence relation on regular expressions. A complete axiomatization for Kleene's theory of 'regular events' and 'regular expressions' was first given by Salomaa in [Sal66]. (However, Salomaa mentions that an almost identical axiomatization was given, together with a completeness proof, by Anderaa in [And65].)[5] The use of coinduction for proving the equivalence of regular expressions has been studied by Rutten in [Rut98], based on the 'differential calculus of events' (as developed, for instance, by Conway in [Con71, Ch. 5]). Following [Rut98] it is not difficult to give a coinductively motivated proof system for the equivalence relation on regular expressions, a system that is conceptually analogous to the axiomatizations of recursive type equality and recursive subtyping given by Brandt and Henglein. Furthermore, a syntactic-matching system can be defined by following a remark in [Rut98] that the coinduction proof method can also be used to detect if two given regular expressions E_1 and E_2 are not equivalent (namely, by showing that the assumption that there is a bisimulation containing the pair $\langle E_1, E_2 \rangle$ leads to a 'contradiction'). Such a syntactic-matching system for equations between regular expressions, a counterpart to the systems $\mathbf{AK}^{=}$ and $\mathbf{AK}_{\mathbf{0}}^{=}$ studied here, can be used for determining whether or not two regular expressions E and F are equivalent by testing for whether or not the equation $E = F$ is consistent relative to the system.

It is very likely that most of the transformations that we have developed for proof systems regarding recursive type equality have counterparts in transformations between respective proof systems for the equivalence relation on regular expressions. What we are able to report as a fact is that the transformation between $\mathbf{HB}_{\mathbf{0}}^{=}$ and $\mathbf{AC}^{=}$ described in Section 8.1 can be adapted for the purpose of establishing an analogous link between similar kinds of proof systems concerning the equivalence relation between regular expressions. Namely, there exists an effective transformation from derivations in an 'analytic' version $\mathbf{cREG_0}$ of a coinductively motivated proof system **cREG** and derivations in a slight variant **REG**, taken from Conway's book [Con71], of the axiomatization by Salomaa.

[5] Salomaa even acknowledges in [Sal66] that some parts of his completeness proof "were essentially simplified by Anderaa's proof"; but in the same passage he goes on to say that Anderaa uses a result "which is not quite correct" (and Salomaa explains the problem in an earlier remark).

For explaining the most important features of the mentioned proof systems in a little more detail, we introduce some basic notions concerning regular expressions. All of our stipulations below presuppose a given finite alphabet Σ that can be chosen according to a specifically considered situation. We do not always formulate it explicitly that the statements we give hold with respect to all choices of finite alphabets Σ, but we hint at such assertions by relativizing all of our designations for sets, functions, and proof systems on regular expressions to the symbol Σ that denotes the alphabet on which the respective sets, functions, and proof systems depend.

We let the set $\mathcal{R}eg(\Sigma)$ of *regular expressions* over alphabet Σ be defined as those words over Σ that are generated by the grammar

$$E \;::=\; 0 \,|\, 1 \,|\, \underbrace{a}_{\in\Sigma} \,|\, E+F \,|\, EF \,|\, E^*$$

and we let the function $\mathcal{L} : \mathcal{R}eg(\Sigma) \to \mathcal{P}(\Sigma^*)$ that to every regular expression E assigns the formal language $\mathcal{L}(E) \subseteq \Sigma^*$ *represented by* E be defined inductively by the clauses

$$\begin{array}{ccc} \mathcal{L}(0) = \emptyset \;, & \mathcal{L}(1) = \{\varepsilon\} \;, & \mathcal{L}(a) = \{a\} \;\; (\text{f.a. } a \in \Sigma) \;, \\ \mathcal{L}(E+F) = \mathcal{L}(E) \cup \mathcal{L}(F) \;, & \mathcal{L}(EF) = \mathcal{L}(E)\mathcal{L}(F) \;, & \mathcal{L}(E^*) = \mathcal{L}(E)^* \;, \end{array}$$

where the 'concatenation product' $L_1 L_2$ of two formal languages L_1 and L_2, and the 'iteration' L^* of a language L are defined

$$\begin{aligned} L_1 L_2 \;&=_{\text{def}}\; \{w_1 w_2 \,|\, w_1 \in L_1,\, w_2 \in L_2\} \;, \\ L^* \;&=_{\text{def}}\; \bigcup_{n\in\omega} L^n \;, \qquad \text{where} \qquad \begin{aligned} L^0 \;&=_{\text{def}}\; \{\epsilon\} \;, \quad \text{and} \\ L^{i+1} \;&=_{\text{def}}\; LL^i \quad (\text{for all } i \in \omega) \;. \end{aligned} \end{aligned}$$

Two regular expressions E and F are called *equivalent* (denoted by $E =_{\mathcal{L}} F$) if and only if $\mathcal{L}(E) = \mathcal{L}(F)$ holds, i.e. iff they represent the same formal language.

Furthermore a function $o : \mathcal{R}eg(\Sigma) \to \{0,1\}$, $E \mapsto o(E)$ is defined by

$$o(E) \;=_{\text{def}}\; \begin{cases} 1 & \ldots \; \epsilon \in \mathcal{L}(E) \\ 0 & \ldots \; \epsilon \notin \mathcal{L}(E) \;. \end{cases}$$

It is easy to give a definition of o using induction on the length of regular expressions. In the terminology of Salomaa, a regular expression $E \in \mathcal{R}eg(\Sigma)$ has the *empty word property* if and only if $o(E) = 1$ holds.

Based on the 'differential calculus of events' (see [Con71, Ch. 5]), it is furthermore easy to give, for all $a \in \Sigma$, inductive definitions for *a-derivatives* on $\mathcal{R}eg(\Sigma)$, that is, for functions

$$(\cdot)_a : \mathcal{R}eg(\Sigma) \to \mathcal{R}eg(\Sigma),\; E \mapsto E_a$$

that have the respective property

$$\mathcal{L}(E_a) = \{w \,|\, aw \in \mathcal{L}(E)\} \quad (\text{for all } a \in \Sigma \text{ and } E \in \mathcal{R}eg(\Sigma)) \;.$$

(The a-derivative E_a of a regular expression E, for all $E \in \mathcal{R}eg(\Sigma)$ and $a \in \Sigma$, represents the "a-derivative" of the language represented by E, where, for all $a \in \Sigma$, the *a-derivative* L_a of a language L over Σ is defined by $L_a =_{\text{def}} \{w \mid aw \in L\}$ in [Con71].) In an inductive definition of $(\cdot)_a$, for arbitrary $a \in \Sigma$, the clauses concerning regular expressions with $+$ and * as their outermost symbols can respectively be chosen as

$$(E+F)_a =_{\text{def}} E_a + F_a \qquad\qquad (E^*)_a =_{\text{def}} E_a E^*$$

(the inductive clause for regular expressions with the concatenation symbol as outermost symbol is slightly more involved).

The characteristic feature of the axiomatization $\mathbf{F_1}$ given by Salomaa in [Sal66] is a *fixed-point rule* which facilitates that "regular expression equations of the form $E = EF + G$ are solvable by a rule of inference" ([Sal66])[6]. This rule is schematically defined by

$$\frac{E = EF + G}{E = GF^*} \ (\text{if } o(F) = 0). \tag{9.5}$$

A different version of this fixed-point rule is used by Conway in [Con71] for a system that differs from the axiomatizations $\mathbf{F_1}$ and $\mathbf{F_2}$ in [Sal66] also in a number of axioms, despite the fact that it is introduced under the name "Salomaa's axiomatization". Applications of the fixed-point rule used in [Con71] result from inference figures in (9.5) by replacing products EF and GF^* by FE and F^*G, respectively; this rule is schematically defined by

$$\frac{E = FE + G}{E = F^*G} \ \text{CONTRACT} \ (\text{if } o(F) = 0) \tag{9.6}$$

(we have decided to call it CONTRACT here in order to underscore its similarity with the fixed-point rule bearing the same name in the proof system $\mathbf{AC}^{=}_{*}$). The reason behind Conway's choice of this version of the fixed-point rule (9.5) in the Salomaa system $\mathbf{F_1}$ seems to be that it is much closer related to the particular differential calculus of events he introduces (in which the derivative L_a of a language L is defined as the language of all words w such that aw (instead of wa) is a word of L). We designate Conway's system by $\mathbf{REG}(\Sigma)$: more precisely, we let $\mathbf{REG}(\Sigma)$ be the (pure) Hilbert system that contains all axioms referred to in [Con71, p.107] (namely, the axioms (C1)–(C13) given in [Con71, p.25]), the rule CONTRACT defined as in (9.6), and additionally all rules of equational logic for equations between regular expressions over alphabet Σ (the rules for reflexivity, symmetry, transitivity, substitution, and context-compatibility); Conway only mentions the rule (9.6), but other rules for reasoning with equations are obviously needed as well[7].

A coinductively motivated proof system $\mathbf{cREG}(\Sigma)$ for the equivalence relation on regular expressions with respect to alphabet $\Sigma = \{a_1, \ldots, a_n\}$ can be built by

[6] We have used E, F, and G here instead of the syntactical variables α, β and γ that are employed for regular expressions throughout [Sal66].

[7] In the case of Salomaa's axiomatization $\mathbf{F_1}$ the rules of equational logic are admissible, as stated by Lemma 1 in [Sal66], owing to the presence of the specific substitution rule (R1) in $\mathbf{F_1}$.

using a ‘circular version’ of the *composition rule*

$$\frac{E_{a_1} = F_{a_1} \qquad \ldots \qquad E_{a_n} = F_{a_n}}{E = F}\;\text{COMP} \quad (\text{if } o(E) = o(F))$$

the soundness of which (with respect to $=_{\mathcal{L}}$) is an easy consequence of the fact

$$(\forall E \in \mathcal{R}eg(\Sigma))\,\big[\,E \;=_{\mathcal{L}}\; o(E) + a_1 E_{a_1} + \ldots + a_n E_{a_n}\,\big]\;; \tag{9.7}$$

(9.7) follows, in its turn, from the definition of the derivatives $(\cdot)_a$ and from the “fundamental theorem of formal languages” (which is a version of (9.7) for languages instead of for regular expressions). In a natural-deduction system formulation, the circular version of COMP is a rule COMP/FIX that allows applications of the form

$$\frac{\begin{array}{c}[E = F]^u\\ \mathcal{D}_1\\ E_{a_1} = F_{a_1}\end{array} \qquad \ldots \qquad \begin{array}{c}[E = F]^u\\ \mathcal{D}_n\\ E_{a_n} = F_{a_n}\end{array}}{E = F}\;\begin{array}{l}\text{COMP/FIX}, u\\ \quad(\text{if } o(E) = o(F)).\end{array}$$

We propose an ‘analytic’ version $\mathbf{cREG_0}(\Sigma)$ of a natural-deduction system named $\mathbf{cREG}(\Sigma)$ that contains the rule COMP/FIX: the system $\mathbf{cREG_0}$ possesses all equations of the form $E = E$ as reflexivity axioms belonging to the scheme called (REFL), it contains COMP/FIX, and furthermore the two rules

$$\frac{\begin{array}{c}\mathcal{D}_1\\ C[E_1] = F\end{array}}{C[E_2] = F}\;\text{App}_l\text{Ax}_{\mathbf{REG}} \qquad\qquad \frac{\begin{array}{c}\mathcal{D}_1\\ F = C[E_1]\end{array}}{F = C[E_2]}\;\text{App}_r\text{Ax}_{\mathbf{REG}(\Sigma)} \tag{9.8}$$

(respectively if $E_1 = E_2$ or $E_2 = E_1$ is an axiom of $\mathbf{REG}(\Sigma)$)

(here C denotes a context for regular expressions with one or more occurrences of a hole $[\,]$, and where, for all such contexts C and $E \in \mathcal{R}eg(\Sigma)$, $C[E]$ stands for the result of hole-filling in C by E). The two rules in (9.8) allow to apply axioms of Conway’s system $\mathbf{REG}(\Sigma)$ in subexpressions of regular expressions on the left-, or respectively, on the right-hand side of an equation. The system $\mathbf{cREG_0}(\Sigma)$ does not contain symmetry and transitivity rules and it can therefore be viewed to be ‘analytic’ (in a certain precise formulation of this notion).

Now we are able to give a rough sketch of the connection mentioned earlier between the systems $\mathbf{cREG_0}$ and $\mathbf{REG}$: there exists an effective transformation from $\mathbf{cREG_0}(\Sigma)$-derivations into $\mathbf{REG}(\Sigma)$-derivations that proceeds by analogous steps as the transformation from $\mathbf{HB_0^=}$-derivations into $\mathbf{AC^=}$-derivations developed in Section 8.1. More precisely, there exists an annotated version $\textbf{ann-cREG}_\mathbf{0}(\Sigma)$ of the system $\mathbf{cREG_0}(\Sigma)$ such that an arbitrary derivation $\mathcal{D}$ in $\mathbf{cREG_0}(\Sigma)$ without assumptions and with conclusion $E = F$, for some $E, F \in \mathcal{R}eg(\Sigma)$, can be transformed into a derivation $\mathcal{D}'$ in $\mathbf{REG}(\Sigma)$ without assumptions and with the same conclusion as $\mathcal{D}$ by successively performing the three steps described in the items below.

(1) *Annotation Step*: The derivation $\mathcal{D}$ in $\mathbf{cREG_0}(\Sigma)$ is annotated with the result of a derivation $\hat{\mathcal{D}}$ in the annotated version $\textbf{ann-cREG}_\mathbf{0}(\Sigma)$ of $\mathbf{cREG_0}(\Sigma)$ such that $\hat{\mathcal{D}}$ has conclusion $G : E = F$, for some $G \in \mathcal{R}eg(\Sigma)$, and does not contain open assumptions.

(2) *Extraction Step*: From the derivation $\hat{\mathcal{D}}$ in $\textbf{ann-cREG}_\mathbf{0}(\Sigma)$ two derivations $(\hat{\mathcal{D}})^{(1)}$ and $(\hat{\mathcal{D}})^{(2)}$ in $\mathbf{REG}(\Sigma)$ are ‘extracted’ that do not contain assumptions and that have the respective conclusions $E = G$ and $F = G$.

(3) *Combination Step*: The two derivations $(\hat{\mathcal{D}})^{(1)}$ and $(\hat{\mathcal{D}})^{(2)}$ in $\mathbf{REG}(\Sigma)$ are combined, using one application of each of the rules SYMM and TRANS, into a derivation $\mathcal{D}'$ in $\mathbf{REG}(\Sigma)$ without assumptions and with the same conclusion $E = F$ as that of the initial derivation $\mathcal{D}$.

For this transformation an illustration analogous to Figure 8.1 on page 259 can be given. Some further details of this transformation are available on the slides [Gra04a] of a recent talk.

As mentioned above, it is very likely that also for most of the other transformations that we have found between proof systems for recursive types there exist counterparts in transformations between respective proof systems for regular expressions. But the case of the transformation that has been outlined here has been the only one which we have examined so far.

Appendix A

Proofs of Statements in Chapter 3

In this appendix we collect some of the more technical proofs for statements in Chapter 3. It contains proofs of statements in the sections about substitution expressions involving recursive types (Section 3.3), about the variant relation on recursive types (Section 3.4), about the notions of tree unfolding and leading symbol of a recursive type (Section 3.5), about the relation "weak recursive type equivalence" (Section 3.7), and about the notion of generated subterm of a recursive type (Section 3.9).

A.1 Proofs of Statements in Section 3.3: Substitution Expressions

Proof of Lemma 3.3.11. We will treat the assertions (i) and (ii) of the lemma in the below item (2) and will first consider assertion (iii) in item (1).

(1) The assertion in item (iii) of the lemma is an easy consequence of the one in item (i). Let $\tau, \sigma_1, \sigma_2 \in \mu Tp$ and $\alpha, \beta \in TVar$ be such that $\alpha \not\equiv \beta$ and $\beta \notin \mathrm{fv}(\tau)$. Suppose that the substitution expressions $\mathbf{s}_1 \equiv \tau[\sigma_1/\alpha]\,[\sigma_2/\beta]$ and $\mathbf{s}_2 \equiv \tau[\sigma_2/\beta]\,[\sigma_1[\sigma_2/\beta]/\alpha]$ are both admissible. Then clearly also $\sigma_1[\sigma_2/\beta]$ must be admissible. Hence the assertion in item (i) of the lemma is applicable and it implies that the recursive types denoted by $\mathbf{s}_1$ and $\mathbf{s}_2$ are syntactically equal.

(2) Both of the two statements (i) and (ii) of the lemma can be shown, for all $\tau, \sigma_1, \sigma_2 \in \mu Tp$ and $\alpha, \beta \in TVar$ such that $\alpha \not\equiv \beta$ and $\alpha \notin \mathrm{fv}(\sigma_2)$, by induction on the syntactical depth $|\tau|$ of τ.

For the base cases $|\tau| = 0$ in both inductive proofs the three possible subcases $\tau \equiv \alpha$, $\tau \equiv \beta$ and $\tau \not\equiv \alpha \,\wedge\, \tau \not\equiv \beta$ can be distinguished and treated separately. In each of these subcases the verification of the base case in the respective induction is easy. As an example we consider the subcase $\tau \equiv \beta$: Let $\sigma_1, \sigma_2 \in \mu Tp$ and $\alpha, \beta \in TVar$ be such that $\alpha \not\equiv \beta$ and $\alpha \notin \mathrm{fv}(\sigma_2)$ holds and let $\tau \equiv \beta$. We furthermore suppose that $\sigma_1[\sigma_2/\beta]$ is admissible. Then we find that

$$\left.\begin{aligned} \tau[\sigma_1/\alpha]\,[\sigma_2/\beta] \;&\Lleftarrow\!\!\!\Rrightarrow\; \beta[\sigma_1/\alpha]\,[\sigma_2/\beta] \;\Lleftarrow\!\!\!\Rrightarrow\; \beta[\sigma_2/\beta] \;\Lleftarrow\!\!\!\Rrightarrow\; \sigma_2 \\ &\underset{\text{(I)}}{\Lleftarrow\!\!\!\Rrightarrow}\; \sigma_2[\sigma_1[\sigma_2/\beta]/\alpha] \;\Lleftarrow\!\!\!\Rrightarrow\; \beta[\sigma_2/\beta]\,[\sigma_1[\sigma_2/\beta]/\alpha] \;\Lleftarrow\!\!\!\Rrightarrow \\ &\Lleftarrow\!\!\!\Rrightarrow\; \tau[\sigma_2/\beta]\,[\sigma_1[\sigma_2/\beta]/\alpha] \end{aligned}\right\} \tag{A.1}$$

where for “$\Rrightarrow$” in the equality equivalence (I) the admissibility of $\sigma_1[\sigma_2/\beta]$ entered and where “$\Lleftarrow$” in (I) is due to the assumption $\alpha \notin \mathrm{fv}(\sigma_2)$. By reading the above chain of equality equivalences as a sequence of equality implications from left to right we find a demonstration for the base case $|\tau| = 0$ in its subcase $\tau \equiv \beta$ for an inductive proof of (i). On the other hand the assumption, that $\sigma_1[\sigma_2/\beta]$ is admissible, is not needed as justification for any of the equality implications passed in (A.1) on the way from right to left. Thus (A.1) is also able to settle the subcase $\tau \equiv \beta$ of the base case in an inductive proof for assertion (ii) of the lemma.

In the induction step of both inductive proofs the cases $\tau \equiv \tau_1 \to \tau_2$ for some $\tau_1, \tau_2 \in \mu Tp$ and $\tau \equiv \mu\gamma.\,\tau_0$ for some $\tau_0 \in \mu Tp$ and $\gamma \in TVar$ can be distinguished; moreover it is useful to divide the second case again into the three subcases $\gamma \equiv \alpha$, $\gamma \equiv \beta$ and $\gamma \not\equiv \alpha \,\wedge\, \gamma \not\equiv \beta$. Performing the induction step consists always of detailed though largely similar considerations. We will establish the induction steps only for the somewhat interesting subcase $\gamma \equiv \beta$, i.e. that $\tau \equiv \mu\beta.\,\tau_0$ holds for some $\tau_0 \in \mu Tp$. It will turn out that the induction hypotheses is not needed in this subcase[1] and that therefore the assertions in (i) and (ii) will be proved directly in this situation.

To show the induction step for an inductive proof of (i) we let $\tau_0, \sigma_1, \sigma_2 \in \mu Tp$, $\alpha, \beta \in TVar$ such that $\alpha \not\equiv \beta$, $\alpha \notin \mathrm{fv}(\sigma_2)$. We set $\tau \equiv \mu\beta.\,\tau_0$, assume that $\sigma_1[\sigma_2/\beta]$ is admissible and we furthermore suppose that $\mathbf{s}_1 \equiv \tau[\sigma_1/\alpha]\,[\sigma_2/\beta]$ is admissible. Then we find that

$$\left.\begin{aligned} \underbrace{\tau[\sigma_1/\alpha]\,[\sigma_2/\beta]}_{\equiv\,\mathbf{s}_1} \;&\Lleftarrow\!\!\!\Rrightarrow\; (\mu\beta.\,\tau_0)\,[\sigma_1/\alpha]\,[\sigma_2/\beta] \;\underset{\text{(I)}}{\Rrightarrow} \\ &\underset{\text{(I)}}{\Rrightarrow}\; (\mu\beta.\,\tau_0[\sigma_1/\alpha])\,[\sigma_2/\beta] \;\Lleftarrow\!\!\!\Rrightarrow \\ &\Lleftarrow\!\!\!\Rrightarrow\; \mu\beta.\,\tau_0[\sigma_1/\alpha] \;\underset{\text{(II)}}{\Rrightarrow}\; (\mu\beta.\,\tau_0)\,[\sigma_1/\alpha] \;\Rrightarrow \\ &\underset{\text{(III)}}{\Rrightarrow}\; (\mu\beta.\,\tau_0)\,[\sigma_1[\sigma_2/\beta]/\alpha] \;\Lleftarrow\!\!\!\Rrightarrow\; \underbrace{\tau[\sigma_2/\beta]\,[\sigma_1[\sigma_2/\beta]/\alpha]}_{\equiv\,\mathbf{s}_2} \end{aligned}\right\} \tag{A.2}$$

[1] It turns out that in the respective inductive proofs the induction hypothesis is only needed for performing the induction steps in the subcase $\gamma \not\equiv \alpha \wedge \gamma \not\equiv \beta$ of the case $\tau \equiv \mu\gamma.\,\tau_0$ (for some $\gamma \in TVar$ and $\tau_0 \in TVar$) as well as in the case $\tau \equiv \tau_1 \to \tau_2$ (for some $\tau_1, \tau_2 \in \mu Tp$).

where the equality implications (I), (II) and (III) can be justified as follows: (I) and (II) come from applications of Lemma 3.3.10, (i) and (ii), where for (II) it is also used that $\alpha \notin \mathrm{fv}(\tau_0) \vee \beta \notin \mathrm{fv}(\sigma_1)$, which is implied by the admissibility of $(\mu\beta.\,\tau_0)[\sigma_1/\alpha]$ as subexpression of $\mathbf{s}_1$ which is admissible by assumption. For (III) it is used that $\beta \notin \mathrm{fv}(\sigma_1) \vee (\alpha \notin \mathrm{fv}(\mu\beta.\,\tau_0) \wedge$ $\sigma_1[\sigma_2/\beta]$ is admissible) holds here, which (because $\sigma_1[\sigma_2/\beta]$ is admissible) follows if $\alpha \in \mathrm{fv}(\mu\beta.\,\tau_0) \Rightarrow \beta \notin \mathrm{fv}(\sigma_1)$ is demonstrated: but this holds due to the fact that if $\alpha \in \mathrm{fv}(\mu\beta.\,\tau_0)$, then (since $(\mu\beta.\,\tau_0)[\sigma_1/\alpha]$ is admissible as a part of $\mathbf{s}_1$, which is admissible by assumption) the variable β cannot occur free in σ_1. – Hence the chain of equality implications from left to right in (A.2) shows (because we have supposed that $\mathbf{s}_1$ is admissible) that $\mathbf{s}_2$ is admissible and that furthermore the recursive types denoted by $\mathbf{s}_1$ and $\mathbf{s}_2$ are the same. This is what needed to be shown for the induction step in the here considered subcase for an inductive proof of (i).

To show the induction step for an inductive proof of (ii), we let $\tau_0, \sigma_1, \sigma_2 \in$ $\in \mu Tp$, $\alpha, \beta \in TVar$ such that $\alpha \not\equiv \beta$, $\alpha \notin \mathrm{fv}(\sigma_2)$ and we set $\tau \equiv \mu\beta.\,\tau_0$. Furthermore we assume that the hypotheses ($\beta \notin \mathrm{fv}(\sigma_1) \vee \tau[\beta/\sigma]$ is admissible) of the assertion (ii) of the lemma is fulfilled and we let $\mathbf{s}_2 \equiv \mu\beta.\,\tau_0[\sigma_1[\sigma_2/\beta]/\alpha]$. Then we find first that

$$\underbrace{\tau[\sigma_2/\beta]\,[\sigma_1[\sigma_2/\beta]/\alpha]}_{\equiv \mathbf{s}_2} \Leftrightarrow (\mu\beta.\,\tau_0)\,[\sigma_1[\sigma_2/\beta]/\alpha] \underset{\text{(I)}}{\Rrightarrow} \mu\beta.\,\tau_0[\sigma_1[\sigma_2/\beta]/\alpha] \qquad (A.3)$$

holds, where (I) follows from Lemma 3.3.10, (i). For the remaining argument we distinguish—according to our assumption above—the two possible cases that either $\beta \notin \mathrm{fv}(\sigma_1)$ holds or that $\tau[\beta/\sigma]$ is admissible: If $\beta \notin \mathrm{fv}(\sigma_1)$, then we observe that

$$\left.\begin{aligned} &\mu\beta.\,\tau_0[\sigma_1[\sigma_2/\beta]/\alpha] \Leftrightarrow \mu\beta.\,\tau_0[\sigma_1/\alpha] \Leftrightarrow \\ &\qquad \Leftrightarrow (\mu\beta.\,\tau_0[\sigma_1/\alpha])\,[\sigma_2/\beta] \underset{\text{(II)}}{\Rrightarrow} \tau[\sigma_1/\alpha]\,[\sigma_2/\beta] \end{aligned}\right\} \qquad (A.4)$$

holds, where (II) follows from Lemma 3.3.10, (ii), and $\beta \notin \mathrm{fv}(\sigma_1)$, the hypothesis here. If on the other hand $\tau[\beta/\alpha]$ is admissible, then it follows because of $\tau \equiv \mu\beta.\,\tau_0$ that $\alpha \notin \mathrm{fv}(\tau_0)$ and $\alpha \notin \mathrm{fv}(\tau)$, and we find that

$$\left.\begin{aligned} &\mu\beta.\,\tau_0[\sigma_1[\sigma_2/\beta]/\alpha] \Leftrightarrow \mu\beta.\,\tau_0 \Leftrightarrow \\ &\qquad \Leftrightarrow (\mu\beta.\,\tau_0)\,[\sigma_2/\beta] \Leftrightarrow \tau[\sigma_1/\alpha]\,[\sigma_2/\beta] \end{aligned}\right\} \qquad (A.5)$$

is the case. Let us now assume that $\mathbf{s}_2$ is admissible. Then due to (A.3), (A.4) and (A.5) and the fact that (A.4) and (A.5) apply to complementary situations we can conclude that $\mathbf{s}_1 \equiv \tau[\sigma_1/\alpha]\,[\sigma_2/\beta]$ is admissible and that the recursive types denoted by $\mathbf{s}_1$ and $\mathbf{s}_2$ are the same. – By this argument we have now successfully performed the induction step for the inductive proof of (ii) in the subcase $\tau \equiv \mu\beta.\,\tau_0$ (for some $\tau_0 \in \mu Tp$) considered here.

□

A.2 Proof of a Statement in Section 3.4: Variant Relation

Proof of statement (3.18) *in Lemma 3.4.2.* For this, we have to show that, for all $\tau, \tau', \sigma \in \mu Tp$ and $\alpha \in TVar$,

$$\tau \equiv_{\text{ren}} \tau' \implies \tau[\sigma/\alpha] \equiv_{\text{ren}} \tau'[\sigma/\alpha] \tag{A.6}$$

holds. We develop the proof in the three items below.

(i) First it will be shown that the following holds:

$$\begin{aligned}&(\forall \tau, \tau', \sigma \in \mu Tp)\,(\forall \alpha \in TVar)\\ &\qquad \big[\, \tau \rightarrow_{\text{ren}} \tau' \implies \tau[\sigma/\alpha] \rightarrow_{\text{ren}} \tau'[\sigma/\alpha]\,\big]\ .\end{aligned} \tag{A.7}$$

Perhaps a warning is in order in connection with this statement as well as with the statement of the lemma: spelling out the implicit side-condition used for occurrences of substitutions this assertion reads, more explicitly, that only if σ is substitutible for α in both τ and τ' there must exist a reduction step $\tau[\sigma/\alpha] \rightarrow_{\text{ren}} \tau'[\sigma/\alpha]$ given that $\alpha \in TVar$ and $\tau, \tau', \sigma \in \mu Tp$ such that there is a reduction step $\tau \rightarrow_{\text{ren}} \tau'$.

For showing (A.7), we let arbitrary $\tau, \tau', \sigma \in \mu Tp$ and $\alpha \in TVar$ be given with the properties that $\tau \rightarrow_{\text{ren}} \tau'$ holds and that σ is substitutible for α in both τ and τ'.

Then $\tau \equiv \chi[\mu\epsilon.\,\rho/\delta]$ for some $\chi, \rho \in \mu Tp$ and some $\epsilon, \delta \in TVar$, and $\tau' \equiv$ $\equiv \chi[\mu\epsilon'.\,\rho[\epsilon'/\epsilon]]$ for some variable $\epsilon' \notin \text{fv}(\rho)$ such that ϵ' is substitutible in ρ for ϵ; w.l.o.g. here χ can be assumed to have been chosen such that $\delta \notin \text{fv}(\sigma)$.

If $\alpha \notin \text{fv}(\mu\epsilon.\,\rho)$, then $\alpha \notin \text{fv}(\mu\epsilon'.\,\rho[\epsilon'/\epsilon])$ holds. Thus $\tau[\sigma/\alpha] \equiv \chi[\sigma/\alpha][\mu\epsilon.\,\rho/\delta]$ and $\tau'[\sigma/\alpha] \equiv \chi[\sigma/\alpha][\mu\epsilon'.\,\rho[\epsilon'/\epsilon]/\delta]$ follow, which shows that $\tau[\sigma/\alpha] \rightarrow_{\text{ren}} \tau'[\sigma/\alpha]$ holds.

If $\alpha \in \text{fv}(\mu\epsilon.\,\rho)$, then also $\alpha \in \text{fv}(\mu\epsilon'.\,\rho[\epsilon'/\epsilon])$ holds. Then $\tau[\sigma/\alpha]$ is equal to $\chi[\sigma/\alpha][\mu\epsilon.\,\rho[\sigma/\alpha]/\delta]$ and because $\epsilon, \epsilon' \notin \text{fv}(\sigma)$ (which follows from $\alpha \in \text{fv}(\mu\epsilon.\,\rho)$ as well as from the fact, that σ is substitutible for α in both τ and τ') $\tau'[\sigma/\alpha]$ $\equiv \chi[\sigma/\alpha][\mu\epsilon'.\,\rho[\epsilon'/\epsilon][\sigma/\alpha]/\delta] \equiv \chi[\sigma/\alpha][\mu\epsilon'.\,\rho[\sigma/\alpha][\epsilon'/\epsilon]/\delta]$, hence also in this case $\tau[\sigma/\alpha] \rightarrow_{\text{ren}} \tau'[\sigma/\alpha]$.

(ii) For given $\alpha \in TVar$ and $\tau, \tau', \sigma \in \mu Tp$ with $\tau \equiv_{\text{ren}} \tau'$, the assertion (A.6) follows directly from (i) in case that there exists a finite chain $\langle \tau_0, \tau_1, \ldots, \tau_n \rangle$ (for some $n \in \omega\backslash\{0\}$) of recursive types, such that $\tau \equiv \tau_0 \rightarrow_{\text{ren}} \tau_1 \rightarrow_{\text{ren}} \cdots \rightarrow_{\text{ren}} \tau_n$ and $\tau_n \equiv \tau'$ (such a chain exists because of $\tau \equiv_{\text{ren}} \tau'$ due to the definition of $\equiv_{\text{ren}}$) and σ is substitutible for α in each term τ_i $(0 \leq i \leq n)$: then clearly $\tau_i[\sigma/\alpha] \rightarrow_{\text{ren}} \tau_{i+1}[\sigma/\alpha]$ for all $i = 0, 1, \ldots, n-1$ follows from (i), which leads to the chain $\tau[\sigma/\alpha] \equiv \tau_0[\sigma/\alpha] \rightarrow_{\text{ren}} \tau_1[\sigma/\alpha] \rightarrow_{\text{ren}} \cdots \rightarrow_{\text{ren}} \tau_n[\sigma/\alpha] \equiv \tau'[\sigma/\alpha]$ and thus gives $\tau[\sigma/\alpha] \equiv_{\text{ren}} \tau'[\sigma/\alpha]$.

However, if σ is substitutible for α in τ and τ', this does not automatically also mean that it is substitutible for α in each type belonging to an *arbitrary* given $\rightarrow_{\text{ren}}$-chain between the variants τ and τ'. It remains to show, that "good" $\rightarrow_{\text{ren}}$-chains do always exist (i.e. such chains, for which the argumentation here can take place). In the next item it will be shown (a proof for this will be sketched) that an arbitrary $\rightarrow_{\text{ren}}$-chain can moreover always be transformed effectively into a "good" one of the same length.

(iii) If for some $n \in \omega\backslash\{0\}$ and $\tau_0, \tau_1, \ldots, \tau_n \in \mu Tp$ in a given $\rightarrow_{\text{ren}}$-chain $\tau_0 \rightarrow_{\text{ren}} \tau_1 \rightarrow_{\text{ren}} \ldots \rightarrow_{\text{ren}} \tau_n$ the recursive type σ is substitutible for α in τ_0 and τ_n, then there exists also an $\rightarrow_{\text{ren}}$-chain $\tau'_0 \rightarrow_{\text{ren}} \tau'_1 \rightarrow_{\text{ren}} \ldots \rightarrow_{\text{ren}} \tau'_n$ with $\tau_0 \equiv \tau'_0$ and $\tau_n \equiv \tau'_n$, such that σ is substitutible for α in each τ_i $(0 \leq i \leq n)$. This can be shown as follows.

For given α and σ this can be shown by induction on the number of terms τ_i in a given chain $\tau_0 \rightarrow_{\text{ren}} \ldots \rightarrow_{\text{ren}} \tau_n$, in which σ is *not* substitutible for α.

Let α and σ be arbitrary, but fixed in the following.

If this number is zero, nothing remains to be shown. If it is greater then zero, let us assume that α is not substitutible for σ already in τ_1 in a given chain $\tau_0 \rightarrow_{\text{ren}} \tau_1 \rightarrow_{\text{ren}} \ldots \rightarrow_{\text{ren}} \tau_n$, where σ is substitutible for α in τ_0 and τ_n; if this happens only later in this chain, then it can be argued analogously. The situation here means that, for $\chi, \rho \in \mu Tp$ and $\epsilon, \tilde{\epsilon} \in \mathit{TVar}$, $\tau_0 \equiv \chi[\mu\epsilon.\, \rho/\delta]$, $\alpha \in \text{fv}(\mu\epsilon.\, \rho)$, $\tau_1 \equiv \chi[\mu\tilde{\epsilon}.\, \rho[\tilde{\epsilon}/\epsilon]/\delta]$ with $\tilde{\epsilon} \notin \text{fv}(\rho)$, $\tilde{\epsilon}$ is substitutible for ϵ in ρ and $\tilde{\epsilon} \in \text{fv}(\sigma)$ (since σ is substitutible for α in τ_0 but not in τ_1).

If it is now set $\tau'_0 \equiv \tau_0$, $\tau'_1 \equiv \chi[\mu\tilde{\tilde{\epsilon}}.\, \rho[\tilde{\tilde{\epsilon}}/\epsilon]/\delta]$ for a (completely new) variable $\tilde{\tilde{\epsilon}}$, where $\tilde{\tilde{\epsilon}} \notin \text{fv}(\rho)$, $\tilde{\tilde{\epsilon}}$ substitutible for α in ρ, $\tilde{\tilde{\epsilon}} \notin \text{fv}(\sigma)$ and $\tilde{\tilde{\epsilon}}$ does not occur in *any* of the τ_i $(0 \leq i \leq n)$, then σ is substitutible for α in τ'_1 and $\tau'_0 \rightarrow_{\text{ren}} \tau'_1$. This $\rightarrow_{\text{ren}}$-reduction can furthermore be extended to a $\rightarrow_{\text{ren}}$-chain $\tau_0 \equiv \tau'_0 \rightarrow_{\text{ren}} \tau'_1 \rightarrow_{\text{ren}} \ldots \rightarrow_{\text{ren}} \tau'_n \equiv \tau_n$ of length n between τ_0 and τ_n: Since $\tilde{\tilde{\epsilon}}$ does not occur in the original chain at all, it is clear that all renamings of bound variables, that happen in $\tau_1 \rightarrow_{\text{ren}} \tau_2 \rightarrow_{\text{ren}} \tau_3 \rightarrow_{\text{ren}} \ldots \rightarrow_{\text{ren}} \tau_n$ can then be carried out similarly in a $\rightarrow_{\text{ren}}$-chain starting from τ'_1 (except that when for the first time after the initial step in the original chain in a variant $\mu\tilde{\epsilon}.\, (\rho[\tilde{\epsilon}/\epsilon])'$ of $\mu\tilde{\epsilon}.\, \rho[\tilde{\epsilon}/\epsilon]$ the variable $\tilde{\epsilon}$ (occurring in one τ_i at the same position as $\mu\tilde{\epsilon}.\, \rho[\tilde{\epsilon}/\epsilon]$ in τ_1) is renamed to some other variable $\tilde{\epsilon}^*$, now the new bound variable $\tilde{\tilde{\epsilon}}$ in a corresponding variant $\mu\tilde{\tilde{\epsilon}}.\, (\rho[\tilde{\tilde{\epsilon}}/\epsilon])'$ of $\mu\tilde{\tilde{\epsilon}}.\, \rho[\tilde{\tilde{\epsilon}}/\epsilon]$ will be renamed to $\tilde{\epsilon}^*$).

σ is then substitutible in τ'_1 of the chain $\tau_0 \equiv \tau'_0 \rightarrow_{\text{ren}} \tau'_1 \rightarrow_{\text{ren}} \ldots \rightarrow_{\text{ren}} \tau'_n \equiv \tau_n$ and also in all those τ'_i, for which σ was already substitutible for α in τ_i $(0 \leq i \leq n)$. Thus the number of recursive types τ'_i in the new chain, in which σ is not substitutible for α, has decreased at least by one, which allows to apply the induction hypothesis.

□

A.3 Proofs of Statements in Section 3.5: Tree Unfolding and Leading Symbol

Proof of Lemma 3.5.7. We let $\tau, \tau' \in \mu Tp$ and $\alpha_1 \in TVar$ be arbitrary such that $\tau' \equiv_{\text{ren}} \tau$ and such that $\mu\alpha_1.\,\tau$ is substitutible for α_1 in τ. For showing (3.29), we have to prove the following three implications:

$$\text{nl}\mu\text{b}(\,\tau'[\mu\alpha_1.\,\tau/\alpha_1]\,) < \text{nl}\mu\text{b}(\mu\alpha_1.\,\tau) \quad \underset{\text{(I)}}{\Longrightarrow} \tag{A.8}$$

$$\left.\begin{array}{l} \underset{\text{(I)}}{\Longrightarrow} \quad \neg\,(\exists\, n \in \omega\backslash\{0\})\,(\exists\, \alpha_2, \ldots, \alpha_n \in TVar) \\ \qquad\qquad\qquad [\,\tau \equiv \mu\alpha_2 \ldots \alpha_n.\,\alpha_1 \;\;\&\;\; \alpha_1 \not\equiv \alpha_2, \ldots, \alpha_n\,] \end{array}\right\} \tag{A.9}$$

$$\underset{\text{(II)}}{\Longrightarrow} \quad \text{nl}\mu\text{b}(\,\tau'[\mu\alpha_1.\,\tau/\alpha_1]\,) = \text{nl}\mu\text{b}(\mu\alpha_1.\,\tau) - 1 \tag{A.10}$$

$$\underset{\text{(III)}}{\Longrightarrow} \quad \text{nl}\mu\text{b}(\,\tau'[\mu\alpha_1.\,\tau/\alpha_1]\,) < \text{nl}\mu\text{b}(\mu\alpha_1.\,\tau)\ .$$

We demonstrate these three implications in the three items below.

"$\underset{\text{(I)}}{\Rightarrow}$": We reason indirectly and assume $\neg$ (A.9) holds. Then there exist $n \in \omega\backslash\{0\}$ and $\alpha_2, \ldots, \alpha_n \in TVar$ such that $\tau \equiv \mu\alpha_2 \ldots \alpha_n.\,\alpha_1$ and $\alpha_1 \not\equiv \alpha_2, \ldots, \alpha_n$ holds; we choose n and $\alpha_2, \ldots, \alpha_n$ in this way. Then $\tau' \equiv \mu\tilde{\alpha}_2 \ldots \tilde{\alpha}_n.\,\alpha_1$ holds for some $\tilde{\alpha}_2, \ldots, \tilde{\alpha}_n \in TVar$ with $\alpha_1 \not\equiv \tilde{\alpha}_2, \ldots, \tilde{\alpha}_n$. But now we find:

$$\begin{aligned} \text{nl}\mu\text{b}(\,\tau'[\mu\alpha_1.\,\tau/\alpha_1]\,) \;&=\; \text{nl}\mu\text{b}(\,\mu\tilde{\alpha}_2 \ldots \tilde{\alpha}_n\alpha_1 \ldots \alpha_n.\,\alpha_1\,) \;= \\ &=\; 2n-1 \;\geq\; n \;=\; \text{nl}\mu\text{b}(\mu\alpha_1.\,\tau)\ . \end{aligned}$$

Hence we have inferred $\neg$ (A.8).

"$\underset{\text{(II)}}{\Rightarrow}$": We let $n = \text{nl}\mu\text{b}(\mu\alpha_1.\,\tau)$, which entails $n \geq 1$, and we assume that (A.9) holds.

Due to the choice of n, we find that $\tau \equiv \mu\alpha_2 \ldots \alpha_n.\,\tau_0$ for some $\alpha_2, \ldots, \alpha_n \in$ $\in TVar$ and a recursive type τ_0 that does not have a leading μ-binding. From our assumption on α_1 and τ we can now conclude that either $\tau_0 \equiv \alpha_1$ with $\alpha_1 \in \{\alpha_2, \ldots, \alpha_n\}$ or that $\tau_0 \in \{\bot, \top\} \cup (TVar \setminus \{\alpha_1\})$ or that $\tau_0 \equiv \rho_1 \to \rho_2$ for some $\rho_1, \rho_2 \in \mu Tp$ is the case. This implies that if the variant τ' of τ is written as $\tau' \equiv \mu\tilde{\alpha}_2 \ldots \tilde{\alpha}_n.\,\tilde{\tau}_0$ with appropriate $\tilde{\alpha}_2, \ldots, \tilde{\alpha}_n \in TVar$ and with some $\tilde{\tau}_0 \in \mu Tp$ that does not have a leading μ-binding, then it holds that: Either $\tilde{\tau}_0 \equiv \tilde{\alpha}_k$ for some $k \in \{2, \ldots, n\}$, or $\tilde{\tau}_0 \in \{\bot, \top\} \cup (TVar \setminus \{\alpha_1\})$ holds, or $\tilde{\tau}_0$ is equal to $\tilde{\rho}_1 \to \tilde{\rho}_2$ for some $\tilde{\rho}_1, \tilde{\rho}_2 \in \mu Tp$.

In the first two cases we find that $\alpha_1 \notin \text{fv}(\tau')$ and hence

$$\text{nl}\mu\text{b}(\,\tau'[\mu\alpha_1.\,\tau/\alpha_1]\,) \;=\; \text{nl}\mu\text{b}(\tau') \;=\; n-1 \;=\; \text{nl}\mu\text{b}(\mu\alpha_1.\,\tau) - 1\ .$$

And in the third case we confirm

$$\begin{aligned} \text{nl}\mu\text{b}(\,\tau'[\mu\alpha_1.\,\tau/\alpha_1]\,) \;&=\; \text{nl}\mu\text{b}(\,\mu\tilde{\alpha}_2 \ldots \tilde{\alpha}_n.\,(\tilde{\rho}_1[\mu\alpha_1.\,\tau/\alpha_1] \to \tilde{\rho}_2[\mu\alpha_1.\,\tau/\alpha_1])\,) \\ &=\; n-1 \;=\; \text{nl}\mu\text{b}(\mu\alpha_1.\,\tau) - 1\ . \end{aligned}$$

In all three cases we have thus shown (A.10).

“$\underset{\text{(III)}}{\Rightarrow}$”: This implication is obvious.

□

Proof for Lemma 3.5.10. We demonstrate the statements (i) and (ii) of the lemma in the below two items (a) and (b), respectively.

(a) This can also be shown through a proof by an induction of the same kind as used for the purpose of Definition 3.5.8, in which for the treatment of case (5) the assertion is relied on that

$$\begin{aligned}\mu\alpha.\,\tau_0 \equiv_{\text{ren}} \mu\tilde{\alpha}.\,\tilde{\tau}_0 \;\;\&\;\; \tau_0' \equiv_{\text{ren}} \tau_0 \;\;\&\;\; \tilde{\tau_0}' \equiv_{\text{ren}} \tilde{\tau}_0 \;\;\Longrightarrow&\\ \Longrightarrow\;\; \tau_0'\,[\mu\alpha.\,\tau_0/\alpha] \equiv_{\text{ren}} \tilde{\tau}_0'\,[\mu\tilde{\alpha}.\,\tilde{\tau}_0/\tilde{\alpha}]&\end{aligned} \tag{A.11}$$

holds for all $\tau_0, \tau_0', \tilde{\tau}_0, \tilde{\tau}_0' \in \mu Tp$ and $\alpha, \tilde{\alpha} \in \mu Tp$; hereby the auxiliary statement (A.11) is an easy consequence of those assertions of Lemma 3.4.2 that are associated with (3.16), (3.19), and (3.24).

(b) We let $\alpha \in TVar$ and $\tau_0, \tau_0' \in \mu Tp$ be arbitrary such that $\tau_0' \equiv_{\text{ren}} \tau_0$ and $\mu\alpha.\,\tau_0$ is substitutible for α in τ_0. We distinguish the following three cases:

Case 1: $\alpha \notin \text{fv}(\tau_0)$.
It follows that $\alpha \notin \text{fv}(\tau_0')$. Here we are in case (3) of Definition 3.5.8 and can readily confirm

$$\textbf{Tree}(\mu\alpha.\,\tau_0) = \textbf{Tree}(\tau_0) = \textbf{Tree}(\tau_0'\,[\mu\alpha.\,\tau_0/\alpha]) \;.$$

Case 2: $\mu\alpha.\,\tau_0 \equiv \mu\alpha\alpha_2\ldots\alpha_n.\,\alpha$ for some $\alpha_2,\ldots,\alpha_n \in TVar$ such that $\alpha \not\equiv$ $\not\equiv \alpha_2,\ldots,\alpha_n$.
Then it follows that $\tau_0' \equiv \mu\tilde{\alpha}_2\ldots\tilde{\alpha}_n.\,\alpha$ for some $\tilde{\alpha}_2,\ldots,\tilde{\alpha}_n \in TVar$ with the property $\alpha \not\equiv \tilde{\alpha}_2,\ldots,\tilde{\alpha}_n$. Here we are in case (4) of Definition 3.5.8 and find that

$$\begin{aligned}\textbf{Tree}(\mu\alpha.\,\tau) = \textbf{Tree}(\,\mu\alpha\alpha_2\ldots\alpha_n.\,\alpha\,) = \;\ldots\; =&\\ = \textbf{Tree}(\,\mu\tilde{\alpha}_1\ldots\tilde{\alpha}_n\alpha\alpha_2\ldots\alpha_n.\,\alpha\,) = \textbf{Tree}(\,\tau_0'\,[\mu\alpha.\,\tau_0/\alpha]\,) \;.&\end{aligned}$$

As justification for the second until $(n+1)$-st equality sign (from left to right) we have hereby used the definition of **Tree** in case (3) of Definition 3.5.8 (because of $\text{fv}(\mu\alpha\alpha_2\ldots\alpha_n.\,\alpha) = \emptyset$ and the obvious fact that, for all $i \in \{2,\ldots,n\}$, $\text{fv}(\mu\tilde{\alpha}_i\ldots\tilde{\alpha}_n\alpha\alpha_2\ldots\alpha_n.\,\alpha) = \emptyset$ holds).

Case 3: $\mu\alpha.\,\tau_0 \equiv \mu\alpha\alpha_2\ldots\alpha_n.\,(\rho_1 \to \rho_2)$ for some $\alpha_2,\ldots,\alpha_n \in TVar$ with the property $\alpha \not\equiv \alpha_2,\ldots,\alpha_n$ and for some $\rho_1, \rho_2 \in \mu Tp$ with $\alpha \in \text{fv}(\tau_1 \to \tau_2)$.
Here we are in case (5) of Definition 3.5.8 and find that

$$\textbf{Tree}(\mu\alpha.\,\tau_0) = \textbf{Tree}(\tau_0''\,[\mu\alpha.\,\tau_0/\alpha]) = \textbf{Tree}(\tau_0'\,[\mu\alpha.\,\tau_0/\alpha]) \;,$$

where τ_0'' is the appropriate variant of τ_0 chosen by (the algorithm underlying) case (5) of Definition 3.5.8. In the second equality we have used Lemma 3.4.2, (3.18).

□

A.4 Proofs of Statements in Section 3.7: Weak Recursive Type Equivalence

Proof of Lemma 3.7.5. The statements (i) and (ii) of the proposition are considered separately in the two items below.

(a) That CTXT is a derivable rule of **WEQ** can be shown by straightforward induction on the depth $|C|$ of the involved context $C \in \mu Tp\text{–}Ctxt$. That SUBST is an admissible rule in **WEQ** follows from the *proof*[2] of Lemma 7.1.9, which states that SUBST is admissible in a proper extension of **WEQ**, the system $\mathbf{AC}^{=}$ defined in Chapter 5.

(b) We first want to exhibit a short proof that relies on a result from [Gra03a], and then we give a slightly longer proof that uses statements from Chapter 4 (and that gives a hint of how the statement from [Gra03a] used in the first case can be proved). Both arguments start from the observation that, as a consequence of assertion (i) of the proposition and of Lemma 4.2.4, (i), the rules CTXT and SUBST are both admissible in **WEQ**.

First proof. From this it follows that all axioms and all rules of **EQL** are admissible in **WEQ** (in the notation of [Gra03a], that $\mathbf{EQL} \preceq_{r/adm} \mathbf{WEQ}$ holds). And this implies, by Theorem 4.12 in [Gra03a, p.23], that **WEQ** is an extension of **EQL** (which there is abbreviated by $\mathbf{EQL} \preceq_{th} \mathbf{WEQ}$).

Second proof. For showing that **WEQ** is an extension of **EQL**, it suffices to demonstrate that, for every derivation $\mathcal{D}$ in **EQL** without assumptions, there exists a mimicking derivation $\mathcal{D}'$ in **WEQ**. To prove this, let $\mathcal{D}$ be an arbitrary derivation in **EQL** without assumptions; we show the existence of a mimicking derivation in **WEQ**. Since all axioms and rules of **EQL** except SUBST and CTXT are also axioms and rules of **WEQ**, it follows that $\mathcal{D}$ is a derivation without assumptions in **WEQ**+SUBST+CTXT. As an easy consequence of Theorem 4.2.8, (i), the facts that SUBST and CTXT are admissible in **WEQ** imply that SUBST-elimination and CTXT-elimination hold in **WEQ**+SUBST+CTXT. And from this it follows easily that a derivation $\mathcal{D}'$ in **WEQ** without assumptions and with the same conclusion as $\mathcal{D}$ exists.[3]

□

Proof of Lemma 3.7.9. The axiom system $\mathbf{AC}^{=}$ for $=_\mu$ due to Amadio and Cardelli, which is defined in Definition 5.1.1, Chapter 5, is an extension of the system **WEQ** (indeed, $\mathbf{AC}^{=}$ is an extension by enlargement of **WEQ** by adding the single rule UFP). Therefore the soundness of **WEQ** with respect to $=_\mu$ follows from the soundness theorem for $\mathbf{AC}^{=}$ with respect to $=_\mu$, Theorem 5.1.4, which is proved in [AmCa93]. □

[2] However, it does not follow from the assertion of Lemma 7.1.9 itself.

[3] It is actually not difficult to define a process for SUBST- and CTXT-elimination for derivations without assumptions in **WEQ**+SUBST+CTXT according to which $\mathcal{D}'$ can always be found effectively from $\mathcal{D}$.

A.5 Proofs of Statements in Section 3.9: Generated Subterms

In this appendix we gather the proofs for some technical lemmas that we have formulated, and used, in Section 3.9, Chapter 3, on generated subterms of recursive types.

We start with giving the proof of Lemma 3.9.9, which asserts in its item (i) that free variables of a generated subterm of a recursive type σ are also free variables of σ, and in its item (ii) that all free variables of a recursive type σ are generated subterms of σ.

Proof of Lemma 3.9.9. The items (i) and (ii) are treated in the respective items (1) and (2) below.

(1) It is easy to check that for all rule applications $\dfrac{\tau \sqsubseteq \sigma_0}{\tau \sqsubseteq \sigma_1}\,R$ of **gST** it holds that $\mathrm{fv}(\sigma_0) \subseteq \mathrm{fv}(\sigma_1)$. By using this, the assertion "$\mathrm{fv}(\tau) \subseteq \mathrm{fv}(\sigma)$ for arbitrary given $\tau, \sigma \in \mu Tp$ such that τ is a generated subterm of σ" can be proved by straightforward induction on the depth of derivations in **gST**.

(2) For given α this item can be proved for all $\sigma \in \mu Tp$ by induction on pairs of the form $(\mathrm{m}\mu\mathrm{d}_\alpha(\sigma), |\sigma|)$ in $\omega \times \omega$ with respect to the lexicographic ordering on this set; hereby $\mathrm{m}\mu\mathrm{d}_\alpha(\sigma)$ denotes the *minimal μ-depth of a free occurrence of α in σ*, i.e. the minimal number of μ-operators in the term-structure of σ above a free occurrence of α in σ. More precisely, for all $\tilde{\sigma} \in \mu Tp$ the positive integer $\mathrm{m}\mu\mathrm{d}_\alpha(\tilde{\sigma})$ is inductively defined as follows:

$$\mathrm{m}\mu\mathrm{d}_\alpha(\tilde{\sigma}) =_{\mathrm{def}}$$
$$=_{\mathrm{def}} \begin{cases} 0 & \ldots\ \alpha \notin \mathrm{fv}(\tilde{\sigma}) \text{ or } \tilde{\sigma} \equiv \alpha \\ \min\big\{\mathrm{m}\mu\mathrm{d}_\alpha(\tilde{\sigma}_i) \mid i \in \{1,2\}, & \\ \qquad\qquad\qquad \alpha \in \mathrm{fv}(\tilde{\sigma}_i)\big\} & \ldots\ \alpha \in \mathrm{fv}(\tilde{\sigma}) \text{ and } \tilde{\sigma} \equiv \tilde{\sigma}_1 \to \tilde{\sigma}_2 \\ \mathrm{m}\mu\mathrm{d}_\alpha(\tilde{\sigma}_0) + 1 & \ldots\ \alpha \in \mathrm{fv}(\tilde{\sigma}) \text{ and } \tilde{\sigma} \equiv \mu\beta.\,\tilde{\sigma}_0 \end{cases}$$

In one case in the proof below, we will use the following facts about $\mathrm{m}\mu\mathrm{d}_\alpha(\cdot)$: For all $\tilde{\sigma}_0, \tilde{\sigma}_1, \tilde{\sigma}_2 \in \mu Tp$ and $\alpha, \beta, \beta' \in TVar$ it holds, that

$$\tilde{\sigma}_1 \equiv_{\mathrm{ren}} \tilde{\sigma}_2 \implies \mathrm{m}\mu\mathrm{d}_\alpha(\tilde{\sigma}_1) = \mathrm{m}\mu\mathrm{d}_\alpha(\tilde{\sigma}_2)\ , \tag{A.12}$$
$$\alpha \in \mathrm{fv}(\tilde{\sigma}_0),\ \alpha \not\equiv \beta \implies \mathrm{m}\mu\mathrm{d}_\alpha(\tilde{\sigma}_0[\mu\beta.\,\tilde{\sigma}_0/\beta]) < \mathrm{m}\mu\mathrm{d}_\alpha(\mu\beta.\,\tilde{\sigma}_0)\ . \tag{A.13}$$

Here (A.12) follows by induction on the length of a $\to_{\mathrm{ren}}$-chain between $\tilde{\sigma}_1$ and $\tilde{\sigma}_2$ using the following additional properties:

$$\begin{aligned} \beta, \beta' \not\equiv \alpha &\implies \mathrm{m}\mu\mathrm{d}_\alpha(\tilde{\sigma}[\beta'/\beta]) = \mathrm{m}\mu\mathrm{d}_\alpha(\tilde{\sigma}) \\ \beta' \notin \mathrm{fv}(\tilde{\sigma}_0) &\implies \mathrm{m}\mu\mathrm{d}_\alpha(\mu\beta.\,\tilde{\sigma}_0) = \mathrm{m}\mu\mathrm{d}_\alpha(\mu\beta'.\,(\tilde{\sigma}_0[\beta'/\beta])) \\ \mathrm{m}\mu\mathrm{d}_\alpha(\tilde{\sigma}_1) = \mathrm{m}\mu\mathrm{d}_\alpha(\tilde{\sigma}_2) &\implies \mathrm{m}\mu\mathrm{d}_\alpha(\tilde{\tau}[\tilde{\sigma}_1/\beta]) = \mathrm{m}\mu\mathrm{d}_\alpha(\tilde{\tau}[\tilde{\sigma}_2/\beta]) \end{aligned}$$

(for all $\tilde{\tau}, \tilde{\sigma}, \tilde{\sigma}_0, \tilde{\sigma}_1, \tilde{\sigma}_2 \in \mu Tp$ and $\alpha, \beta, \beta' \in TVar$) which can be shown by easy inductions. On the other hand the reason for (A.13) in somewhat intuitive argumentation consists in the following: Clearly for each free occurrence of α in $\tilde{\sigma}_0[\mu\beta.\,\tilde{\sigma}_0/\beta]$ within a subexpression $\mu\beta.\,\tilde{\sigma}_0$ there exists another free occurrence of α in $\tilde{\sigma}_0[\mu\beta.\,\tilde{\sigma}_0/\beta]$, which originates from a free occurrence of α in $\tilde{\sigma}_0$ and which is in $\tilde{\sigma}_0[\mu\beta.\,\tilde{\sigma}_0/\beta]$ preceded by at least one μ-binding less (i.e. whose μ-depth is strictly less). Therefore the free occurrences of α in $\tilde{\sigma}_0[\mu\beta.\,\tilde{\sigma}_0/\beta]$ (if there are any) of minimal μ-depth must have μ-depth $\mathrm{m}\mu\mathrm{d}_\alpha(\tilde{\sigma}_0)$. Thus

$$\alpha \not\equiv \beta \implies \mathrm{m}\mu\mathrm{d}_\alpha(\tilde{\sigma}_0[\mu\beta.\,\tilde{\sigma}_0/\beta]) = \mathrm{m}\mu\mathrm{d}_\alpha(\tilde{\sigma}_0) \tag{A.14}$$

holds for all $\tilde{\sigma}_0 \in \mu Tp$ and variables α, β. Since by the definition of $\mathrm{m}\mu\mathrm{d}_\alpha(\cdot)$

$$\alpha \in \mathrm{fv}(\tilde{\sigma}_0),\ \alpha \not\equiv \beta \implies \mathrm{m}\mu\mathrm{d}_\alpha(\tilde{\sigma}_0) < 1 + \mathrm{m}\mu\mathrm{d}_\alpha(\tilde{\sigma}_0) = \mathrm{m}\mu\mathrm{d}_\alpha(\mu\beta.\,\tilde{\sigma}_0)$$

holds for all variables α, β and $\tilde{\sigma}_0 \in \mu Tp$, then (A.13) follows. – A strictly formal proof of (A.14) could e.g. involve the statement, that for all $\tau, \sigma \in \mu Tp$ and all variables α, β such that $\alpha \not\equiv \beta$ variables α, β it holds, that

$$\mathrm{m}\mu\mathrm{d}_\alpha(\tau[\sigma/\beta]) =_{\mathrm{def}}$$

$$=_{\mathrm{def}} \begin{cases} \mathrm{m}\mu\mathrm{d}_\alpha(\tau) & \ldots\ \beta \notin \mathrm{fv}(\tau) \vee \alpha \notin \mathrm{fv}(\sigma) \\ \mathrm{m}\mu\mathrm{d}_\beta(\tau) + \mathrm{m}\mu\mathrm{d}_\alpha(\sigma) & \ldots\ \beta \in \mathrm{fv}(\tau) \wedge \alpha \in \mathrm{fv}(\sigma) \setminus \mathrm{fv}(\tau) \\ \min\{\mathrm{m}\mu\mathrm{d}_\alpha(\tau), & \\ \qquad \mathrm{m}\mu\mathrm{d}_\beta(\tau) + \mathrm{m}\mu\mathrm{d}_\alpha(\sigma)\} & \ldots\ \beta \in \mathrm{fv}(\tau) \wedge \alpha \in \mathrm{fv}(\sigma) \cap \mathrm{fv}(\tau) \end{cases}$$

which can be shown by induction on $|\tau|$.

We proceed now with the induction in the form described above for the statement of item (ii) of the lemma.

For the base case of the induction now let $\mathrm{m}\mu\mathrm{d}_\alpha(\sigma) = 0$ and $|\sigma| = 0$. If $\alpha \in \mathrm{fv}(\sigma)$, then $\sigma \equiv \alpha$, hence $\alpha \equiv \sigma \in \mathcal{G}(\sigma)$ follows (due to the fact, that $\alpha \sqsubseteq \alpha$ is an axiom (REFL) of the axiom system **gST**).

Let now $\mathrm{m}\mu\mathrm{d}_\alpha(\sigma) > 0$. As consequences we have $\alpha \in \mathrm{fv}(\sigma)$ and $|\sigma| > 0$.

If $\sigma \equiv \sigma_1 \to \sigma_2$, then there exists $i \in \{1,2\}$ such that $\mathrm{m}\mu\mathrm{d}_\alpha(\sigma_i) = \mathrm{m}\mu\mathrm{d}_\alpha(\sigma)$ and hence also $\mathrm{m}\mu\mathrm{d}_\alpha(\sigma_i) > 0$. It follows that $\alpha \in \mathrm{fv}(\sigma_i)$ and thus by the induction-hypothesis (which is applicable because of $|\sigma_i| < |\sigma|$) it follows that $\alpha \in \mathcal{G}(\sigma_i)$. Since clearly $\mathcal{G}(\sigma_i) \subseteq \mathcal{G}(\sigma)$ is the case, we find that also $\alpha \in \mathcal{G}(\sigma)$.

Now let $\sigma \equiv \mu\beta.\,\sigma_0$. We choose a variant σ_0' of σ_0 such that σ is substitutible for β in σ_0'. Since $\alpha \in \mathrm{fv}(\sigma) = \mathrm{fv}(\mu\beta.\,\sigma_0) = \mathrm{fv}(\mu\beta.\,\sigma_0')$, it follows with (A.13) and (A.12) that $\mathrm{m}\mu\mathrm{d}_\alpha(\sigma_0'[\mu\beta.\,\sigma_0'/\beta]) < \mathrm{m}\mu\mathrm{d}_\alpha(\mu\beta.\,\sigma_0') = \mathrm{m}\mu\mathrm{d}_\alpha(\sigma)$. Since it is clearly also the case that $\alpha \in \mathrm{fv}(\sigma_0'[\mu\beta.\,\sigma_0'/\beta]) = \mathrm{fv}(\mu\beta.\,\sigma_0)$, we can conclude from the induction hypothesis that $\alpha \in \mathcal{G}(\sigma_0'[\mu\beta.\,\sigma_0'/\beta])$. Since $\sigma_0'[\mu\beta.\,\sigma_0'/\beta]$ clearly is a generated subterm of $\sigma \equiv \mu\beta.\,\sigma_0$, we find here again that $\alpha \in \mathcal{G}(\sigma)$.

This concludes the induction and thus the proof of this item (ii) of the lemma.

□

We continue by developing a proof of Lemma 3.9.10. For this purpose, we first formulate, and then prove, another lemma that consists of a technical statement about derivations in **gST** without assumptions and with conclusions of the form $\tau \sqsubseteq \sigma[\rho/\alpha]$, with some $\tau, \sigma, \rho, \in \mu Tp$ and $\alpha \in TVar$.

Lemma A.5.1. *Let $\sigma, \rho \in \mu Tp$ and $\alpha \in TVar$ such that ρ is substitutible for α in ρ. And furthermore, let $\tilde{\sigma} \in \mu Tp$ be such that $\tilde{\sigma} \equiv_{ren} \sigma[\rho/\alpha]$.*

Then for every derivation

$$\begin{array}{c} \dfrac{\text{(REFL)}}{(\tau \sqsubseteq \tau)} \\ \mathcal{D} \\ \tau \sqsubseteq \tilde{\sigma} \end{array}$$

in **gST** *one of the following two statements is the case:*

(I) For all formulas $\tau \sqsubseteq \omega$ in $\mathcal{D}$ it holds that $\omega \equiv_{ren} \chi[\rho/\alpha]$ with some $\chi \in \mathcal{G}(\sigma)$.

(II) $\mathcal{D}$ is of the form

$$\begin{array}{c} \dfrac{\text{(REFL)}}{(\tau \sqsubseteq \tau)} \\ \mathcal{D}_t \\ (\tau \sqsubseteq \underbrace{\quad \rho' \quad}_{\equiv_{ren} \rho \equiv \alpha[\rho/\alpha]}) \\ \mathcal{D}_b \\ \tau \sqsubseteq \tilde{\sigma} \end{array}$$

where ρ' is a variant of ρ (i.e. $\rho' \equiv_{ren} \rho$ holds), where $\mathcal{D}_t$ is a **gST***-derivation, and where $\mathcal{D}_b$ is a* **gST***-derivation from the assumption $\tau \sqsubseteq \rho'$ for which statement (I) is true, i.e. for all formulas $\tau \sqsubseteq \omega$ in $\mathcal{D}_b$ it holds that $\omega \equiv_{ren} \chi[\rho/\alpha]$ with some $\chi \in \mathcal{G}(\sigma)$.*

Proof. We prove the lemma by induction on the depth $|\mathcal{D}|$ of $\mathcal{D}$ for all derivations $\mathcal{D}$ in **gST** without an assumption and with conclusion $\tau \sqsubseteq \tilde{\sigma}$, where $\tau, \tilde{\sigma} \in \mu Tp$ such that $\tilde{\sigma} \equiv_{\text{ren}} \sigma[\rho/\alpha]$ for some $\sigma, \rho \in \mu Tp$ and $\alpha \in TVar$.

For the base case of the induction, where $|\mathcal{D}| = 0$, clearly assertion (I) of the lemma holds.

Before turning to the induction step, we observe: The following two assertions hold for all $\tilde{\sigma} \in \mu Tp$ with $\tilde{\sigma} \equiv_{\text{ren}} \sigma[\rho/\alpha]$, for some $\sigma, \rho \in \mu Tp$ and $\alpha \in TVar$, and for all derivations $\mathcal{D}$ in **gST** without assumptions and with conclusion $\tau \sqsubseteq \tilde{\sigma}$:

- If $\alpha \notin \text{fv}(\sigma)$, then it follows by Lemma 3.9.9, (i), that $\alpha \notin \text{fv}(\chi)$ for all $\chi \in \mathcal{G}(\sigma)$. As a consequence $\chi[\rho/\alpha] \equiv \chi$ holds for all generated subterms χ of σ, and therefore assertion (I) in the lemma is fulfilled.

- If $\sigma \equiv \alpha$, then $\tilde{\sigma} \equiv_{\mathrm{ren}} \alpha[\rho/\alpha] \equiv \rho$ is the case, and hence for $\mathcal{D}, \tau, \sigma, \rho, \alpha$ assertion (II) in (i) of the lemma holds; this is because of $\tilde{\sigma} \equiv_{\mathrm{ren}} \rho$ the derivation $\mathcal{D}_b$ can be chosen as a trivial derivation in **gST** of depth $|\mathcal{D}_b| = 0$ consisting only of the formula $\tau \sqsubseteq \tilde{\sigma}$ as, at the same time, assumption and conclusion.

As a consequence of these two observations, it suffices to carry out the induction step under the additional requirements $\alpha \in \mathrm{fv}(\sigma)$ and $\sigma \not\equiv \alpha$.

For the induction step, we hence assume $\sigma, \tilde{\sigma}, \rho \in \mu Tp$ and $\alpha \in TVar$ to be given such that $\tilde{\sigma} \equiv_{\mathrm{ren}} \sigma[\rho/\alpha]$ and $\alpha \in \mathrm{fv}(\sigma)$ as well as $\sigma \not\equiv \alpha$ holds, and $\mathcal{D}$ to be a **gST**-derivation without an assumption, with $|\mathcal{D}| > 0$ and with conclusion $\tau \sqsubseteq \tilde{\sigma}$. We distinguish three cases according to which rule of **gST** is applied at the bottom of $\mathcal{D}$.

The cases in which $\mathcal{D}$ ends with an application of either the rule REN_r or the rule $\mathrm{B{\to}CTXT}_r$ are rather straightforward to check and are left out here (the crucial part of the argumentation for these cases resurfaces in the more complicated one treated below).

If the last rule application in $\mathcal{D}$ is a FOLD_r-rule, then $\mathcal{D}$ is of the form

$$\dfrac{\begin{array}{c} \dfrac{\text{(REFL)}}{(\tau \sqsubseteq \tau)} \\ \mathcal{D}_0 \\ \tau \sqsubseteq \tilde{\sigma}_0[\mu\tilde{\beta}.\,\tilde{\sigma}_0/\tilde{\beta}] \end{array}}{\tau \sqsubseteq \mu\tilde{\beta}.\,\tilde{\sigma}_0}\,\mathrm{FOLD}_r$$

where $\mathcal{D}_0$ is again a **gST**-derivation and with $\mu\tilde{\beta}.\,\tilde{\sigma}_0 \equiv \tilde{\sigma}\, (\equiv_{\mathrm{ren}} \sigma[\rho/\alpha])$. Since by assumption $\sigma \not\equiv \alpha$ holds, σ must be of the form $\mu\beta.\,\sigma_0$ for some $\sigma_0 \in \mu Tp$ and $\beta \in TVar$. Due to $\alpha \in \mathrm{fv}(\sigma)$, we get that $\beta \not\equiv \alpha$ and that $\sigma[\rho/\alpha] \equiv (\mu\beta.\,\sigma_0)[\rho/\alpha] \Leftrightarrow$ $\Leftrightarrow \mu\beta.\,\sigma_0[\rho/\alpha]$.

Let furthermore σ_0' be a variant of σ_0 such that ρ is substitutible in σ_0' for α and β and $\mu\beta.\,\sigma_0'$ is substitutible in σ_0' for β. Then Lemma 3.3.13 implies that

$$\sigma_0'[\mu\beta.\,\sigma_0'/\beta][\rho/\alpha] \Leftrightarrow \sigma_0'[\rho/\alpha][\mu\beta.\,\sigma_0'[\rho/\alpha]/\beta] \tag{A.15}$$

holds. And because the substitution expression on the right side of $\Leftrightarrow$ is now admissible, also the substitution expression on the left side follows to be admissible. Because of $\sigma_0 \equiv_{\mathrm{ren}} \sigma_0'$ we also get that $\mu\beta.\,\sigma_0[\rho/\alpha] \equiv_{\mathrm{ren}} \mu\beta.\,\sigma_0'[\rho/\alpha]$. Since $\mu\tilde{\beta}.\,\tilde{\sigma}_0 \equiv_{\mathrm{ren}}$ $\equiv_{\mathrm{ren}} \mu\beta.\,\sigma_0[\rho/\alpha]$, it follows by using Lemma 3.4.2, (3.24), that $\tilde{\sigma}_0[\mu\tilde{\beta}.\,\tilde{\sigma}_0/\tilde{\beta}] \equiv_{\mathrm{ren}}$ $\equiv_{\mathrm{ren}} \sigma_0'[\rho/\alpha][\mu\beta.\,\sigma_0'[\rho/\alpha]/\beta]$ and hence by (A.15) that it is furthermore the case that $\tilde{\sigma}_0[\mu\tilde{\beta}.\,\tilde{\sigma}_0/\tilde{\beta}] \equiv_{\mathrm{ren}} (\sigma_0'[\mu\beta.\,\sigma_0'/\beta])[\rho/\alpha]$.

Thus it is possible to apply the induction hypotheses for $\mathcal{D}_0$ and conclude that one of the following statements $(\mathrm{I})_0$ or $(\mathrm{II})_0$ must hold for $\mathcal{D}_0$:

$(\mathrm{I})_0$ All formulas $\tau \sqsubseteq \omega$ in the derivation $\mathcal{D}_0$ are of the form $\omega \equiv_{\mathrm{ren}} \chi_0[\rho/\alpha]$ for some $\chi_0 \in \mathcal{G}(\sigma_0'[\mu\beta.\,\sigma_0'/\beta])$.

$(\mathrm{II})_0$ $\mathcal{D}_0$ is of the form

$$\begin{array}{c} \dfrac{\text{(REFL)}}{(\tau \sqsubseteq \tau)} \\ \mathcal{D}_t \\ (\tau \sqsubseteq \underbrace{\quad\rho'\quad}_{\equiv_{\text{ren}}\,\rho\,\equiv\,\alpha[\rho/\alpha]}\ ) \\ \mathcal{D}_b \\ \tau \sqsubseteq \tilde{\sigma}_0[\mu\tilde{\beta}.\,\tilde{\sigma}_0/\tilde{\beta}] \end{array}$$

where ρ' is a variant of ρ, $\mathcal{D}_t$ is a **gST**-derivation, and $\mathcal{D}_0$ a **gST**-derivation from assumption $\tau \sqsubseteq \rho'$ and where for all formulas $\tau \sqsubseteq \omega$ of the derivation $\mathcal{D}_b$ it holds that $\omega \equiv_{\text{ren}} \chi_0[\rho/\alpha]$ for some $\chi_0 \in \mathcal{G}(\sigma_0'[\mu\beta.\,\sigma_0'/\beta])$.

Now since obviously every generated subterm of $\sigma_0'[\mu\beta.\,\sigma_0'/\beta]$ is a generated subterm of $\mu\beta.\,\sigma_0'$ and hence also of $\mu\beta.\,\sigma_0$, it follows that $\mathcal{D}$ is of form (I) or (II) dependent on whether $\mathcal{D}_0$ is actually of respective form $(\text{I})_0$ or $(\text{II})_0$.

This concludes the proof of the induction step. Hence we have proved the lemma.

□

Now we can prove Lemma 3.9.10 by demonstrating that it is an easy consequence of Lemma A.5.1.

Proof of Lemma 3.9.10. Let $\sigma, \rho \in \mu Tp$ and $\alpha \in TVar$ be such that ρ is substitutible for α in ρ. The statement

$$\mathcal{G}(\sigma[\rho/\alpha]) \subseteq \{\tau \mid \tau \in \mu Tp,\ \tau \equiv_{\text{ren}} \chi[\rho/\alpha],\ \chi \in \mathcal{G}(\sigma)\} \cup \mathcal{G}(\rho)$$

of Lemma 3.9.10 is now an immediate consequence of the previous lemma: namely, Lemma A.5.1 implies that every generated subterm of $\sigma[\rho/\alpha]$ is either a variant of a recursive type of the form $\chi[\rho/\alpha]$, for some generated subterm χ of σ, or a generated subterm of ρ. □

We set forth by giving a proof for Lemma 3.9.11.

Proof of Lemma 3.9.11. The items (i), (ii) and (iii) of the lemma will respectively be demonstrated in the items (1), (2) and (3) below.

(1) Item (i) follows from the fact that, for all $\sigma \in \{\bot, \top\} \cup TVar$ and $\tau \in \mu Tp$, a formula $\tau \sqsubseteq \sigma$ can only be the conclusion of an application of a rule R in **gST**, if R is a trivial application of REN_r that has the formula $\tau \sqsubseteq \sigma$ also as its premise. Hence for such σ and τ all formulas in a derivation of $\tau \sqsubseteq \sigma$ in $\mathcal{D}$ must be the same and hence it follows that $\tau \equiv \sigma$ because there must be an axiom (REFL) of the form $\tau \sqsubseteq \tau$ at the top of $\mathcal{D}$.

(2) Let $\sigma_1, \sigma_2 \in \mu Tp$.

The set inclusion "$\supseteq$" in (ii) is immediate because clearly all variants of $\sigma_1 \to \sigma_2$ as well as all generated subterms of σ_1 and σ_2 are also generated subterms of $\sigma_1 \to \sigma_2$.

To show the implication "$\subseteq$", let τ be a generated subterm of $\sigma_1 \to \sigma_2$. If $\tau \equiv_{\text{ren}} \sigma_1 \to \sigma_2$, then $\tau \in [\sigma_1 \to \sigma_2]_{\equiv_{\text{ren}}}$ and the desired set inclusion holds obviously. We therefore may assume additionally that $\tau \not\equiv_{\text{ren}} \sigma_1 \to \sigma_2$. Since τ is a generated subterm of $\sigma_1 \to \sigma_2$, there exists a derivation in **gST** with conclusion $\tau \sqsubseteq \sigma_1 \to \sigma_2$; let $\mathcal{D}$ be such a derivation. Our assumption on τ implies that $\mathcal{D}$ must be of the form

$$\begin{array}{cl}
\text{(REFL)} & \\ \hline
(\tau \sqsubseteq \tau) & \\
\mathcal{D}_0 & \\
\tau \sqsubseteq \sigma'_j & \\ \hline
\tau \sqsubseteq \sigma'_1 \to \sigma'_2 & \text{B}{\to}\text{CTXT}_r \\ \hline
\vdots & \text{REN}_r \\ \hline
\tau \sqsubseteq \sigma_1 \to \sigma_2 & \text{REN}_r
\end{array}$$

for some $j \in \{1, 2\}$ and for some $\sigma'_1, \sigma'_2 \in \mu Tp$ with $\sigma'_1 \equiv_{\text{ren}} \sigma_1$ and $\sigma'_2 \equiv_{\text{ren}} \sigma_2$, because not all rule applications in $\mathcal{D}$ can be applications of REN_r and the only other rule application in **gST** with a conclusion of the form $\tau \sqsubseteq \sigma''_1 \to \sigma''_2$ (for general $\sigma''_1, \sigma''_2 \in \mu Tp$; but here only respective variants of σ_1 and σ_2 are possible) is an B$\to$CTXT$_r$-rule. It follows that τ is a generated subterm of σ'_j, and that therefore also τ is a generated subterm of σ_j (since variants of generated subterms always are generated subterms), i.e. $\tau \in \mathcal{G}(\sigma_j)$ holds. Hence τ is also element of the set on the right side of the equation in item (ii) of the lemma. This concludes the proof of the set inclusion "$\subseteq$" in this equation.

(3) For a proof of item (iii), we let a recursive type $\mu\beta.\,\sigma \in \mu Tp$, for some $\sigma \in \mu Tp$ and a type variable β, be given. We also choose a variant σ' of σ with the property that $\mu\beta.\,\sigma$ is substitutible for β in σ'. Then also $\mu\beta.\,\sigma'$ is substitutible for β in σ', and $\mu\beta.\,\sigma' \equiv_{\text{ren}} \mu\beta.\,\sigma$.

We assume that an arbitrary derivation $\mathcal{D}$ in **gST** with conclusion $\tau \sqsubseteq \mu\beta.\,\sigma$, for some $\tau \in \mu Tp$, is given. We have to show that τ is a member of the set on the right side of the set inclusion in item (iii) of the lemma.

This is clearly the case if $\tau \equiv_{\text{ren}} \mu\beta.\,\sigma$. Therefore we now assume additionally that $\tau \not\equiv_{\text{ren}} \mu\beta.\,\sigma$. Let $\tau \sqsubseteq \mu\tilde{\beta}.\,\tilde{\sigma}$ be the topmost occurrence of a formula in $\mathcal{D}$ with a variant of $\mu\beta.\,\sigma$ on the right side of the symbol $\sqsubseteq$ (in particular, we have $\mu\tilde{\beta}.\,\tilde{\sigma} \equiv_{\text{ren}} \mu\beta.\,\sigma$), and let $\mathcal{D}_0$ be the immediate subderivation in $\mathcal{D}$ of that occurrence. This means, that $\mathcal{D}$ is of the form

$$\begin{array}{c}\dfrac{\text{(REFL)}}{(\tau \sqsubseteq \tau)}\\[4pt] \mathcal{D}_0\\[4pt] \tau \sqsubseteq \mu\tilde{\beta}.\,\tilde{\sigma}\\[4pt] \mathcal{D}_b\\[4pt] \tau \sqsubseteq \mu\beta.\,\sigma\end{array}$$

where $\mathcal{D}_b$ is a **gST**-derivation from the assumption $\tau \sqsubseteq \mu\tilde{\beta}.\,\tilde{\sigma}$, and where $\mathcal{D}_0$ is a derivation in **gST** that does not contain a variant of $\mu\beta.\,\sigma$ in formulas at the right side of the symbol $\sqsubseteq$ *except* for the recursive type $\mu\tilde{\beta}.\,\tilde{\sigma}$ in the conclusion of $\mathcal{D}_0$. Since due to our assumption $\tau \not\equiv_{\text{ren}} \mu\beta.\,\sigma$ the conclusion of $\mathcal{D}_0$ cannot be an axiom, and since due to our choice of $\mathcal{D}_0$ there cannot be a REN_r-application the bottom of $\mathcal{D}_0$, the last rule application in $\mathcal{D}_0$ must be that of a FOLD_r-rule. Thus $\mathcal{D}_0$ is of the form

$$\begin{array}{c}\dfrac{\text{(REFL)}}{(\tau \sqsubseteq \tau)}\\[4pt] \mathcal{D}_{00}\\[4pt] \dfrac{\tau \sqsubseteq \tilde{\sigma}[\mu\tilde{\beta}.\,\tilde{\sigma}/\beta]}{\tau \sqsubseteq \mu\tilde{\beta}.\,\tilde{\sigma}}\,\text{FOLD}_r\end{array}$$

where $\mathcal{D}_{00}$ also is a derivation in **gST**. We now go over from $\mathcal{D}_0$ to the derivation $\mathcal{D}'$

$$\begin{array}{c}\dfrac{\text{(REFL)}}{(\tau \sqsubseteq \tau)}\\[4pt] \mathcal{D}_{00}\\[4pt] \dfrac{\dfrac{\dfrac{\tau \sqsubseteq \tilde{\sigma}[\mu\tilde{\beta}.\,\tilde{\sigma}/\beta]}{\tau \sqsubseteq \sigma'[\mu\beta.\,\sigma'/\beta]}\,\text{REN}_r}{\tau \sqsubseteq \mu\beta.\,\sigma'}\,\text{FOLD}_r}{\tau \sqsubseteq \mu\beta.\,\sigma}\,\text{REN}_r\end{array}$$

which has the same conclusion as $\mathcal{D}$. We have used here that

$$\tilde{\sigma}[\mu\tilde{\beta}.\,\tilde{\sigma}/\tilde{\beta}] \equiv_{\text{ren}} \sigma'[\mu\beta.\,\sigma'/\beta] \tag{A.16}$$

holds, which follows from $\mu\tilde{\beta}.\,\tilde{\sigma} \equiv_{\text{ren}} \mu\beta.\,\sigma \equiv_{\text{ren}} \mu\beta.\,\sigma'$ by use of Lemma 3.4.2, (3.24); this statement is applicable here because we have assumed σ' such that $\mu\beta.\,\sigma'$ is substitutible for β in σ' and because we know that $\mu\tilde{\beta}.\,\tilde{\sigma}$ is substitutible for $\tilde{\beta}$ in $\tilde{\sigma}$ from the fact that the application of FOLD_r at the bottom of $\mathcal{D}$ entails this as the implicit side-condition on the substitution expression $\tilde{\sigma}[\mu\tilde{\beta}.\,\sigma/\tilde{\beta}]$ contained in it.

Now let $\mathcal{D}'_{01}$ be the subderivation of $\mathcal{D}'$ that ends at the premise formula $\tau \sqsubseteq \sigma'[\mu\beta.\,\sigma'/\beta]$ of the displayed application of FOLD$_r$. From the way how $\mathcal{D}_0$ was chosen, and from (A.16) we conclude that there does not occur a variant of $\mu\beta.\,\sigma$ at the right side of the symbol $\sqsubseteq$ in formulas $\tau \sqsubseteq \omega$ in $\mathcal{D}'_{01}$. In this situation Lemma A.5.1 implies that $\mathcal{D}'_{01}$ cannot be of respective shape (II) and hence must be of respective form (I) described in that lemma, by which we mean more precisely: for every formula $\tau \sqsubseteq \omega$ in $\mathcal{D}'_{01}$ it holds that $\omega \equiv_{\text{ren}} \chi[\mu\tilde{\beta}.\,\tilde{\sigma}/\tilde{\beta}]$ for some $\chi \in \mathcal{G}(\sigma')$. This must then also hold for the axiom $\tau \sqsubseteq \tau$ at the top of $\mathcal{D}'_{01}$. Hence $\tau \equiv_{\text{ren}} \chi[\mu\beta.\,\sigma'/\beta]$ for some $\chi \in \mathcal{G}(\sigma')$. Since due to $\sigma' \equiv_{\text{ren}} \sigma$ also $\mathcal{G}(\sigma') = \mathcal{G}(\sigma)$ and $\mu\beta.\,\sigma' \equiv_{\text{ren}} \mu\beta.\,\sigma$ hold, it also follows $\tau \equiv_{\text{ren}} \chi[\mu\beta.\,\sigma/\beta]$ for some $\chi \in \mathcal{G}(\sigma)$. This entails that

$$\tau \in \{\omega \mid \omega \equiv_{\text{ren}} \chi[\mu\beta.\,\sigma/\beta], \chi \in \mathcal{G}(\sigma)\}\,.$$

Hence τ is a member of the right side of the inclusion in item (iii) of the lemma.

This concludes the proof of item (iii).

We have completed the proof of the lemma. □

A necessary technical statement for the proof of Lemma 3.9.16 is the following lemma.

Lemma A.5.2. *Let σ be a recursive type for which the variable convention VC holds. Let $\rho, \sigma_1, \ldots, \sigma_n \in \mu Tp$ and let $\beta_1, \ldots, \beta_n \in TVar$ be such that* (3.52) *holds. Then the substitution expression*

$$\rho\,[\mu\beta_n.\,\sigma_n/\beta_n]\ldots[\mu\beta_1.\,\sigma_1/\beta_1] \tag{A.17}$$

is admissible. If, furthermore, $\rho \equiv \mu\beta_{n+1}.\,\sigma_{n+1}$ for some $\beta_{n+1} \in TVar$ and for some $\sigma_{n+1} \in \mu Tp$, then there is a reduction step of $\rightarrow_{\textit{out-unf}}$ of the form

$$\begin{aligned}\rho\,[\mu\beta_n.\,\sigma_n/\beta_n]\ldots[\mu\beta_1.\,\sigma_1/\beta_1] \quad &\rightarrow_{\textit{out-unf}} \\ \rightarrow_{\textit{out-unf}} \quad &\sigma_{n+1}\,[\mu\beta_{n+1}.\,\sigma_{n+1}/\beta_n]\ldots[\mu\beta_1.\,\sigma_1/\beta_1]\end{aligned} \tag{A.18}$$

and in particular, the substitution expression at the right side of (A.18) *is admissible.*

Proof. Let $\sigma \in \mu Tp$ be such that it fulfills the variable condition *VC*, and let $\rho, \sigma_1, \ldots, \sigma_n$ and $\beta_1, \ldots, \beta_n$ be such that (3.52) holds. For showing that (A.17) is admissible, it suffices to prove, for all $i \in \{0, \ldots, n-1\}$, that

$$\rho[\mu\beta_n.\,\sigma_n/\beta_n]\ldots[\mu\beta_i.\,\sigma_i/\beta_i] \tag{A.19}$$

is an admissible substitution expression under the assumption that

$$\rho[\mu\beta_n.\,\sigma_n/\beta_n]\ldots[\mu\beta_{i+1}.\,\sigma_{i+1}/\beta_{i+1}] \tag{A.20}$$

is admissible (this is the induction step in a proof by induction with trivial base case).

For this, we let $i \in \{0, \ldots, n-1\}$ be arbitrary, and we assume that (A.20) is admissible. Due to the way in which $\mu\beta_1.\,\sigma_1, \ldots, \mu\beta_n.\,\sigma_n$ are fixed via (3.52) as nested subterms of σ starting with a μ-binding that contain a considered subterm occurrence of ρ, it follows that

$$\begin{aligned} A &=_{\text{def}} \operatorname{bv}(\rho[\mu\beta_n.\,\sigma_n/\beta_n]\ldots[\mu\beta_{i+1}.\,\sigma_{i+1}/\beta_{i+1}]) \subseteq \operatorname{bv}(\rho) \;\cup\; \{\beta_{i+1}, \ldots, \beta_n\}\;, \\ B &=_{\text{def}} \operatorname{fv}(\mu\beta_i.\,\sigma_i) \subseteq \operatorname{fv}(\sigma) \cup \{\beta_1, \ldots, \beta_{i-1}\}\;. \end{aligned}$$

Due to the variable condition *VC* on σ it follows that the type variables $\beta_1, \ldots, \beta_n$ are distinct, that the bound variables of ρ are different from $\beta_1, \ldots, \beta_n$, and that no bound variable of σ, and hence also none of $\beta_1, \ldots, \beta_n$, can also have a free occurrence in σ. Consequently, the sets A and B have empty intersection. Therefore, replacing $\mu\beta_i.\,\sigma_i$ in the recursive type denoted by (A.20) cannot lead to the capture of variables that are free in $\mu\beta_i.\,\sigma_i$. Hence the substitution expression (A.19) is admissible.

We let again $\sigma \in \mu Tp$ be such that it fulfills the variable condition *VC*, and let $\rho, \sigma_1, \ldots, \sigma_n$ and $\beta_1, \ldots, \beta_n$ be such that (3.52) holds. For showing (A.18), we furthermore assume that $\rho \equiv \mu\beta_{n+1}.\,\sigma_{n+1}$ for some $\beta_{n+1} \in TVar$ and $\sigma_{n+1} \in \mu Tp$. Then we notice the following:

$$\begin{aligned}
&(\mu\beta_{n+1}.\,\sigma_{n+1})[\mu\beta_n.\,\sigma_n/\beta_n]\ldots[\mu\beta_1.\,\sigma_1/\beta_1] \\
&\quad \underset{\text{(I)}}{\Rrightarrow}\; \mu\beta_{n+1}.\,\big(\sigma_{n+1}[\mu\beta_n.\,\sigma_n/\beta_n]\ldots[\mu\beta_1.\,\sigma_1/\beta_1]\big) \\
&\qquad \to_{\text{out-unf}}\; \sigma_{n+1}[\mu\beta_n.\,\sigma_n/\beta_n]\ldots[\mu\beta_1.\,\sigma_1/\beta_1] \\
&\qquad\qquad\qquad \big[\mu\beta_{n+1}.\,\big(\sigma_{n+1}[\mu\beta_n.\,\sigma_n/\beta_n]\ldots[\mu\beta_1.\,\sigma_1/\beta_1]\big)\;/\;\beta_{n+1}\big] \\
&\qquad\qquad \underset{\text{(II)}}{\Rrightarrow}\; \sigma_{n+1}[\mu\beta_n.\,\sigma_n/\beta_n]\ldots[\mu\beta_1.\,\sigma_2/\beta_2][\mu\beta_1.\,\sigma_1/\beta_1] \\
&\qquad\qquad\qquad \big[(\mu\beta_{n+1}.\,\sigma_{n+1})[\mu\beta_n.\,\sigma_n/\beta_n]\ldots \\
&\qquad\qquad\qquad\qquad\qquad \ldots[\mu\beta_2.\,\sigma_2/\beta_2][\mu\beta_1.\,\sigma_1/\beta_1]\;/\;\beta_{n+1}\big] \\
&\qquad\qquad \underset{\text{(III)}}{\Rrightarrow}\; \sigma_{n+1}[\mu\beta_n.\,\sigma_n/\beta_n]\ldots[\mu\beta_2.\,\sigma_2/\beta_2] \\
&\qquad\qquad\qquad \big[(\mu\beta_{n+1}.\,\sigma_{n+1})[\mu\beta_n.\,\sigma_n/\beta_n]\ldots[\mu\beta_2.\,\sigma_2/\beta_2]\;/\;\beta_{n+1}\big] \\
&\qquad\qquad\qquad [\mu\beta_1.\,\sigma_1/\beta_1] \\
&\qquad\qquad \underset{\text{(IV)}}{\Rrightarrow}\; \ldots\ldots \\
&\qquad\qquad \underset{\text{(V)}}{\Rrightarrow}\; \sigma_{n+1}\,[\mu\beta_{n+1}.\,\sigma_{n+1}/\beta_{n+1}]\ldots[\mu\beta_1.\,\sigma_1/\beta_1]\;.
\end{aligned}$$

Hereby the admissibility of all substitution expressions can be seen, similarly as shown above for the first statement of the lemma, to be a consequence of $VC(\sigma)$ and the choice of $\sigma_1, \ldots, \sigma_{n+1}$ and $\alpha_1, \ldots, \alpha_n$. The equality implication (I) follows by repeated application of Lemma 3.3.10, (i), the equality implication (II) follows from Lemma 3.3.10, (ii), and from the admissibility of the substitution expression at the top. The equality implication (III) follows by Lemma 3.3.11, (ii); equally, all further $n-2$ interchanging steps of substitution operations in (IV) and the one in (V) can be justified by Lemma 3.3.11, (ii). In this way we have shown (A.18). □

Remark A.5.3. Closer analysis of this proof shows that we did not use the 'full force' of the variable condition $\mathit{DB}(\sigma)$ as part of the assumption $\mathit{VC}(\sigma)$ on σ. In fact, Lemma A.5.2 also holds under the weaker assumption $\mathit{wVC}(\sigma)$, where the *variable condition wVC* is defined[4], for all $\tilde{\sigma} \in \mu Tp$, by

$$\mathit{wVC}(\tilde{\sigma}) \iff_{\mathrm{def}} \mathit{VC}_0(\tilde{\sigma}) \;\&\; \mathit{wDB}(\tilde{\sigma})\;; \tag{A.21}$$

hereby furthermore the property wDB only demands that μ-binders in nested positions have to bind different type variables (μ-binders in parallel positions may bind the same type variable), i.e. $\mathit{wDB}(\tilde{\sigma})$ is, for all $\tilde{\sigma} \in \mu Tp$, defined as:

$$\neg\,(\exists\, p_1, p_2 \in \mathit{Pos}(\tilde{\sigma}))\,(\exists\, \alpha \in \mathit{TVar})\,(\exists\, \tilde{\sigma}_1, \tilde{\sigma}_2 \in \mu Tp) \\ \big[\, p_1 < p_2 \;\&\; \tilde{\sigma}|_{p_1} = \mu\alpha.\,\tilde{\sigma}_1 \;\&\; \tilde{\sigma}|_{p_2} = \mu\alpha.\,\tilde{\sigma}_2 \,\big]\,. \tag{A.22}$$

The difference with the definition (3.12) of the property DB is that here the condition $p_1 < p_2$ occurs instead of the condition $p_1 \neq p_2$ in (3.12).

Furthermore, it can be checked that even Theorem 3.9.14 holds under the weaker assumption $\mathit{wVC}(\sigma)$ instead of under the assumption $\mathit{VC}(\sigma)$.

Relying on Lemma A.5.2, we can now give a proof for Lemma 3.9.16.

Proof of Lemma 3.9.16. Let σ be a recursive type for which the variable condition VC holds. We have to show the representation

$$\tilde{\mathcal{G}}(\sigma) = \Big\{ \rho\,[\mu\beta_n.\,\sigma_n/\beta_n]\ldots[\mu\beta_1.\,\sigma_1/\beta_1] \mid (3.52) \text{ holds} \Big\} \tag{A.23}$$

for the $\to_{\mathrm{oud}}$-closure of σ (here we have just copied the assertion (3.56) and given it a new equation number). We first notice that the substitution expressions in the set on the right-hand side of the equation in (A.23) are always admissible due to Lemma A.5.2, which guarantees that this expression does indeed denote a well-defined set of recursive types. We abbreviate this set on the right hand side of (A.23) by A, and have to show the two set inclusions "$\subseteq$" and "$\supseteq$" in (A.23), i.e. we show $\tilde{\mathcal{G}}(\sigma) \subseteq A$ and $\tilde{\mathcal{G}}(\sigma) \supseteq A$.

"$\subseteq$": Since $\tilde{\mathcal{G}}(\sigma)$ is the closure of σ under the reduction relation $\to_{\mathrm{oud}}$, to show $\tilde{\mathcal{G}}(\sigma) \subseteq A$ it suffices to demonstrate that $\sigma \in A$, and that A is closed under $\to_{\mathrm{oud}}$. Since $\sigma = \sigma|_\epsilon$, it is easy to see that indeed $\sigma \in A$ holds. Hence it remains to be verified that A is closed under $\to_{\mathrm{oud}}$. For this, we let $\chi_1 \in A$ be arbitrary and assume an arbitrary $\to_{\mathrm{oud}}$-step

$$\chi_1 \to_{\text{out-unf}} \chi_2 \tag{A.24}$$

[4] The variable condition wVC is defined analogously here to the condition for calling a λ-term to be in "weak Barendregt Conventional Form" (wBCF) according to the terminology of a recent paper by R. Vestergaard.

from χ_1, with some $\chi_2 \in \mu Tp$. We have to show that $\chi_2 \in A$. Due to $\chi_1 \in A$ it follows that

$$\chi_1 \;\equiv\; \rho\,[\mu\beta_1.\,\sigma_1/\beta_n]\ldots[\mu\beta_n.\,\sigma_n/\beta_1] \tag{A.25}$$

for some subterm ρ of σ and $\mu\beta_1.\,\sigma_1,\ldots,\mu\beta_n.\,\sigma_n \in \mu Tp$ such that (3.52) holds ($\mu\beta_1.\,\sigma_1,\ldots,\mu\beta_n.\,\sigma_n$ are the nested subterms starting with a μ-expression above a subterm occurrence of ρ in σ).

We distinguish three cases according to which clause in Grammar 3.1 the recursive type ρ was formed in the last step of its formation.

Case 1: $|\rho| = 0$.

If $\rho \equiv \bot$, or $\rho \equiv \top$, or $\rho \equiv \alpha$ for some $\alpha \in TVar$ such that $\alpha \not\equiv \beta_1,\ldots,\beta_n$ holds,

$$\chi_1 \;\equiv\; \rho[\mu\beta_n.\,\sigma_n/\beta_n]\ldots[\mu\beta_1.\,\sigma_1/\beta_1] \;\equiv\; \rho$$

is the case, and then there is actually no $\to_{\text{oud}}$-step $\chi_1 \to_{\text{oud}} \chi_2$ possible, in contradiction with our assumption.

However, if $\rho \equiv \beta_i$ for some $i \in \{1,\ldots,n\}$, then

$$\begin{aligned}\chi_1 \;\equiv\; \rho[\mu\beta_n.\,\sigma_n/\beta_n]\ldots[\mu\beta_1.\,\sigma_1/\beta_1] \;\equiv&\\ \equiv\; \mu\beta_i.\,\sigma_i[\mu\beta_{i-1}.\,\sigma_{i-1}/\beta_{i-1}]\ldots[\mu\beta_1.\,\sigma_1/\beta_1]&\end{aligned}$$

holds because all of the bound variables $\beta_1,\ldots,\beta_n$ of σ are different since they occur in the different, in fact nested, subterms $\mu\beta_1.\,\sigma_1,\ldots,\mu\beta_n.\,\sigma_n$ of the recursive type σ that fulfills the variable condition *VC* by our assumption. It follows that $\chi_1 \in A$ allows also the representation

$$\tilde{\rho}\,[\mu\tilde{\beta}_{\tilde{n}}.\,\tilde{\sigma}_{\tilde{n}}/\tilde{\beta}_{\tilde{n}}]\ldots[\mu\tilde{\beta}_1.\,\tilde{\sigma}_1/\tilde{\beta}_1]$$

with some $\tilde{\rho},\tilde{\sigma}_1,\ldots,\tilde{\sigma}_n \in \mu Tp$ and $\tilde{\beta}_1,\ldots,\tilde{\beta}_n \in TVar$ such that (3.52) holds and where $|\tilde{\rho}| > 0$; obviously we can choose

$$\tilde{n} = i-1, \qquad \tilde{\rho} \equiv \mu\beta_i.\,\sigma_i,$$
$$\mu\tilde{\beta}_1.\,\tilde{\sigma}_1 \equiv \mu\beta_1.\,\sigma_1,\;\;\ldots,\;\; \mu\tilde{\beta}_{\tilde{n}}.\,\tilde{\sigma}_{\tilde{n}} \equiv \mu\beta_i.\,\sigma_i\;.$$

However, this case will be settled below in Case 3.

Case 2: $|\rho| \geq 0$, and $\rho \equiv \rho_1 \to \rho_2$ for some $\rho_1, \rho_2 \in \mu Tp$.

Here it follows

$$\begin{aligned}\chi_1 \;&\equiv\; \rho[\mu\beta_n.\,\sigma_n/\beta_n]\ldots[\mu\beta_1.\,\sigma_1/\beta_1]\\ &\equiv\; \rho_1[\mu\beta_n.\,\sigma_n/\beta_n]\ldots[\mu\beta_1.\,\sigma_1/\beta_1] \to \rho_2[\mu\beta_n.\,\sigma_n/\beta_n]\ldots[\mu\beta_1.\,\sigma_1/\beta_1]\;.\end{aligned}$$

Hence the step $\chi_1 \to_{\text{oud}} \chi_2$ of (A.24), which we consider, can only be a step $\chi_1 \to_{\text{out-dec}} \chi_2$, which implies that

$$\chi_2 \;\equiv\; \rho_i[\mu\beta_n.\,\sigma_n/\beta_n]\ldots[\mu\beta_1.\,\sigma_1/\beta_1]$$

for some $i \in \{1,2\}$. It is easy to check that $\rho_i, \sigma_1, \ldots, \sigma_n$ and $\beta_1, \ldots, \beta_n$ again satisfy (3.52). Hence also $\chi_2 \in A$ follows.

Case 3: $|\rho| \geq 1$, and $\rho \equiv \mu\beta_{n+1}.\, \sigma_{n+1}$ for some $\beta_{n+1} \in \mathit{TVar}$ and for some $\sigma_{n+1} \in \mu Tp$.

Here χ_1, which we assumed to be of the form (A.25) such that (3.52) holds, is of the form

$$\chi_1 \;\equiv\; \mu\beta_{n+1}.\, \sigma_{n+1}[\mu\beta_n.\, \sigma_n/\beta_n] \ldots [\mu\beta_1.\, \sigma_1/\beta_1] \,.$$

Hence the step $\chi_1 \to_{\text{oud}} \chi_2$, which we assumed in (A.24), must be a step of the form $\chi_1 \to_{\text{out-unf}} \chi_2$. Here Lemma A.5.2 implies that such a step is indeed possible and that it has the reduct

$$\chi_2 \equiv \sigma_{n+1}[\mu\beta_{n+1}.\, \sigma_{n+1}/\beta_{n+1}] \ldots [\mu\beta_1.\, \sigma_1/\beta_1]$$

It is now easy to check that for σ_{n+1} and $\sigma_1, \ldots, \sigma_n, \sigma_{n+1}$ and $\beta_1, \ldots, \beta_{n+1}$ again (3.52) holds. This entails $\chi_2 \in A$.

Hence we have shown that in all three possible cases $\chi_2 \in A$ holds. Thus we have demonstrated that A is closed under the relation $\to_{\text{oud}}$.

"$\supseteq$": To show this inclusion, i.e. that $\tilde{\mathcal{G}}(\sigma) \supseteq A$ holds, it has to be proved that indeed every element of A can be reached from σ via a $\to_{\text{oud}}$-reduction sequence. For this, we have to demonstrate that, for all $n \in \omega$, $\sigma, \rho, \sigma_1, \ldots, \sigma_n$ and $\beta_1, \ldots, \beta_n$ the implication

$$(3.52) \quad\Longrightarrow\quad \sigma \twoheadrightarrow_{\text{oud}} \rho[\mu\beta_n.\, \sigma_n/\beta_n] \ldots [\mu\beta_1.\, \sigma_1/\beta_1] \qquad \text{(A.26)}$$

holds. This can be shown by induction on $|\sigma| - |\rho|$.

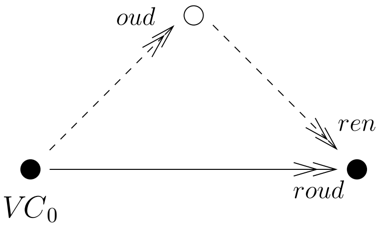
For showing the base case of the induction, we let $|\sigma| - |\rho| = 0$ and we assume (3.52) to hold for $n, \sigma, \rho, \sigma_1, \ldots, \sigma_n$ and $\beta_1, \ldots, \beta_n$. Hence the right hand side of the implication (A.26) follows trivially due to reflexivity of the relation $\twoheadrightarrow_{\text{oud}}$.

For the induction step of the induction, we let $|\sigma| - |\rho| = m + 1 > 0$, for some $m \in \omega$, and we assume that (3.52) holds for some ρ, n, $\sigma_1, \ldots, \sigma_n$ and $\beta_1, \ldots, \beta_n$. We will only treat the case in which ρ is the immediate subterm within σ of a μ-expression $\mu\beta_n.\, \rho$; the cases in which ρ is an immediate subterm within σ of a recursive type $\tilde{\rho} \to \rho$ or $\rho \to \tilde{\rho}$, for some $\tilde{\rho} \in \mu Tp$, are easier to check.

We therefore assume that $\rho \equiv \sigma|_{p_0 1}$ and $\sigma|_{p_0} \equiv \mu\beta_n.\, \sigma_n$ with $\sigma_n \equiv \rho$, and for some $\beta_n \in \mathit{TVar}$ and $p_0 \in \mathit{Acc}(\sigma)$ such that also $p_0 1 \in \mathit{Acc}(\sigma)$. Then it follows that

$$\rho[\mu\beta_n.\, \sigma_n/\beta_n][\mu\beta_{n-1}.\, \sigma_{n-1}/\beta_n] \ldots [\mu\beta_1.\, \sigma_n/\beta_1] \in A \,,$$
$$(\mu\beta_n.\, \sigma_n)[\mu\beta_{n-1}.\, \sigma_{n-1}/\beta_n] \ldots [\mu\beta_1.\, \sigma_n/\beta_1] \in A \,.$$

Figure A.1: Illustration of the statement of Lemma A.5.4: Every $\rightarrow_{\text{roud}}$-reduction sequence starting from a recursive type that fulfills the variable condition VC_0 can be factorized into a $\rightarrow_{\text{oud}}$-reduction sequence and a subsequent sequence of renaming steps.



Since $|\sigma| - |\mu\beta_n.\,\sigma_n| = m < m+1$, it follows by the induction hypothesis that

$$\sigma \;\twoheadrightarrow_{\text{oud}}\; (\mu\beta_n.\,\underbrace{\sigma_n}_{\equiv\rho})[\mu\beta_{n-1}.\,\sigma_{n-1}/\beta_{n-1}]\dots[\mu\beta_1.\,\sigma_n/\beta_1] \qquad \text{(A.27)}$$

By Lemma A.5.2 the possibility to perform the following $\rightarrow_{\text{out-unf}}$-step

$$\begin{aligned}(\mu\beta_n.\,\sigma_n)[\mu\beta_{n-1}.\,\sigma_n/\beta_{n-1}]\dots[\mu\beta_1.\,\sigma_1/\beta_1] \;&\rightarrow_{\text{out-unf}}\\ &\rightarrow_{\text{out-unf}}\; \rho[\mu\beta_n.\,\sigma_n/\beta_n]\dots[\mu\beta_1.\,\sigma_1/\beta_1] \qquad \text{(A.28)}\end{aligned}$$

follows. (A.27) and (A.28) together imply now

$$\sigma \;\twoheadrightarrow_{\text{oud}}\; \rho[\mu\beta_n.\,\sigma_n/\beta_n]\dots[\mu\beta_1.\,\sigma_1/\beta_1]\;,$$

which is what we had to confirm for the induction step in this case.

□

The following lemma will be our main tool in the proof below of Lemma 3.9.17. It states that every $\rightarrow_{\text{roud}}$-reduction sequence that starts out from a recursive type that fulfill the variable condition VC_0 can be factorized into a $\rightarrow_{\text{oud}}$-reduction sequence that is followed by a reduction sequence consisting entirely of $\rightarrow_{\text{ren}}$-steps.

Lemma A.5.4. *Let $\sigma \in \mu Tp$ be such that $VC_0(\sigma)$ holds. In a $\rightarrow_{roud}$-reduction sequence from σ, $\rightarrow_{ren}$-steps can be postponed over $\rightarrow_{oud}$-steps, that is, for all $\tau \in \mu Tp$ it holds*

$$\sigma \twoheadrightarrow_{roud} \tau \;\Longrightarrow\; (\exists\,\chi \in \mu Tp)\,\big[\,\sigma \twoheadrightarrow_{oud} \chi \twoheadrightarrow_{ren} \tau\,\big]\;. \qquad \text{(A.29)}$$

(For an illustration of this statement see Figure A.1). Hence every generated subterm of σ is a variant of a $\rightarrow_{oud}$-generated subterm of σ.

Proof. This lemma is a consequence of Lemma A.5.5, (iv), below which states that in every $\rightarrow_{\mathrm{roud}}$-reduction sequence from a recursive type that fulfills the variable condition VC_0 $\rightarrow_{\mathrm{ren}}$-multisteps can be postponed over $\rightarrow_{\mathrm{oud}}$-steps. (A.29) can be shown, for all $\rightarrow_{\mathrm{roud}}$-reduction sequences Seq from σ to τ, by induction on the sum of, for each $\rightarrow_{\mathrm{oud}}$-step s in Seq, the number of $\rightarrow_{\mathrm{ren}}$-steps to the left of s in Seq, using Lemma A.5.5, (iv), in the induction step. □

With respect to the statement of the following lemma, which is needed to proof the lemma above, we recall that a recursive type σ satisfies the variable condition VC_0, which is symbolically denoted by $\mathit{VC}_0(\sigma)$, if and only if $\mathrm{fv}(\sigma) \cap \mathrm{bv}(\sigma) = \emptyset$ holds, i.e. iff no variable occurs both free and bound in σ.

Lemma A.5.5. *The following four statements hold:*

(i) The variable condition VC_0 is preserved under $\rightarrow_{roud}$-reduction sequences, that is, for all $\sigma, \tau \in \mu\mathit{Tp}$ it holds that

$$\mathit{VC}_0(\sigma) \;\;\&\;\; \sigma \twoheadrightarrow_{oud} \tau \;\;\Longrightarrow\;\; \mathit{VC}_0(\tau)\;.$$

(ii) Let $\sigma \in \mu\mathit{Tp}$ and $\beta \in \mathit{TVar}$ be such that $\mathit{VC}_0(\mu\beta.\,\sigma)$ holds. Then $\sigma[\mu\beta.\,\sigma/\beta]$ is an admissible substitution expression, and hence a $\rightarrow_{out\text{-}unf}$-step, and hence a $\rightarrow_{roud}$-step

$$\mu\beta.\,\sigma \;\rightarrow_{out\text{-}unf}\; \sigma[\mu\beta.\,\sigma/\beta]$$

is possible.

(iii) A $\rightarrow_{ren}$-multistep from a recursive type σ that fulfills the variable condition VC_0 can be postponed over an arbitrary subsequent $\rightarrow_{oud}$-step, i.e. for all $\sigma, \chi, \tau \in \mu\mathit{Tp}$ it holds:

$$\mathit{VC}_0(\sigma) \;\;\&\;\; \sigma \twoheadrightarrow_{ren} \chi \rightarrow_{oud} \tau \quad\Longrightarrow\quad (\exists\, \chi' \in \mu\mathit{Tp})\,\big[\,\sigma \rightarrow_{oud} \chi' \twoheadrightarrow_{ren} \tau\,\big]\;.$$

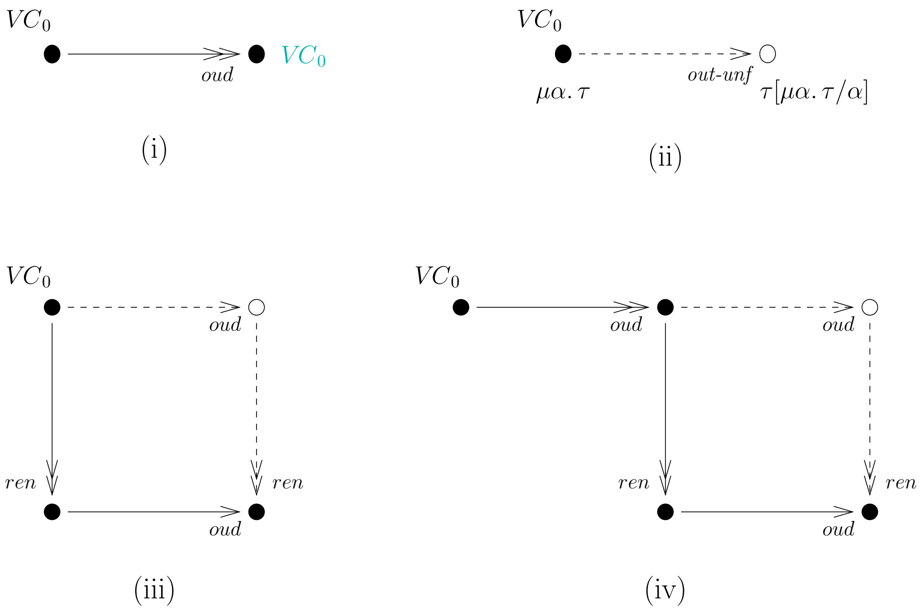
(iv) The first occurring sequence of $\rightarrow_{ren}$-steps in a $\rightarrow_{roud}$-reduction sequence from a recursive type that fulfills the variable condition VC_0 can be postponed over an arbitrary subsequent $\rightarrow_{oud}$-step. That is, for all $\sigma, \chi_1, \chi_2 \in \mu\mathit{Tp}$ it holds that

$$\begin{aligned}\mathit{VC}_0(\sigma) \;\;\&\;\; \sigma \twoheadrightarrow_{oud} \chi_1 \twoheadrightarrow_{ren} \chi_2 \rightarrow_{oud} \tau \;\;\Longrightarrow&\\ \Longrightarrow\;\; (\exists\, \chi_2 \in \mu\mathit{Tp})\,\big[\,\sigma \twoheadrightarrow_{oud} \chi_1 \rightarrow_{oud} \chi_2' \twoheadrightarrow_{ren} \tau\,\big]\;.&\end{aligned}$$

For an illustration of these four statements see Figure A.2.

Sketch of the Proof. (a) (i) follows by induction on the length of $\rightarrow_{\mathrm{oud}}$-reductions sequences by using the fact that VC_0 is preserved by arbitrary $\rightarrow_{\mathrm{oud}}$-steps; this latter fact is easy to verify.

Figure A.2: Illustration of the statements in the items (i), (ii), (iii), and (iv) of Lemma A.5.5:



(i) (ii) (iii) (iv)

(b) If $VC_0(\mu\beta.\,\sigma)$ holds, then no free variable of $\mu\beta.\,\sigma$ can get bound by replacing a free occurrence of β in σ by $\mu\beta.\,\sigma$; hence in this case the $\rightarrow_{\text{out-unf}}$-step

$$\mu\beta.\,\sigma \ \rightarrow_{\text{out-unf}} \ \sigma[\mu\beta.\,\sigma/\beta]$$

is possible.

(c) That $\rightarrow_{\text{out-dec}}$-steps can be postponed over $\rightarrow_{\text{ren}}$-multisteps is easy to see. That also $\rightarrow_{\text{out-unf}}$-steps can be postponed over preceding $\rightarrow_{\text{ren}}$-steps from a recursive type σ that fulfills VC_0 follows from (i) and Lemma 3.4.2, (3.24).

(d) (iv) is a consequence of (iii) and (i).

□

Using the above two lemmas, we can now give the following short proof for Lemma 3.9.17, the second lemma needed in a proof of Theorem 3.9.14.

Proof of Lemma 3.9.17. Let $\sigma \in \mu Tp$ be such that $VC_0(\sigma)$. The inclusion $\tilde{\mathcal{G}}(\sigma) \subseteq$ $\subseteq \mathcal{G}_*(\sigma)$ is an obvious consequence of the fact that the reduction relation $\rightarrow_{\text{oud}}$ is contained in $\rightarrow_{\text{roud}}$ and that hence also $\twoheadrightarrow_{\text{oud}} \subseteq \twoheadrightarrow_{\text{roud}}$ holds.

The inclusion $\mathcal{G}_*(\sigma) \subseteq \tilde{\mathcal{G}}(\sigma)$ follows from Lemma A.5.4 above (that is applicable due to $VC_0(\sigma)$), which implies that every variant of a generated subterm of σ is a variant of a $\rightarrow_{\text{oud}}$-generated subterm of σ. □

And finally, we give the somewhat technical proof for Lemma 3.9.25, which gives a bound for the number of leading μ-bindings of a generated subterm of a recursive type.

Proof of Lemma 3.9.25. For the use in this proof, we define three auxiliary notions: Firstly, we denote, for all $\tau \in \mu Tp$, by $\text{mns}\mu\text{b}(\tau)$ the *maximum number of successive μ-bindings* in τ that is defined by

$$\begin{aligned} &\text{mns}\mu\text{b}(\tau) =_{\text{def}} \\ &\qquad =_{\text{def}} \big\{ n \mid n \in \omega,\, (\exists\, (\mu\alpha_1 \ldots \alpha_n.\, \rho) \in \mu Tp) \,\big[\, \mu\alpha_1 \ldots \alpha_n.\, \rho \trianglelefteq \tau \,\big] \big\} \,. \end{aligned} \tag{A.30}$$

Secondly, we let, for all $\tau \in \mu Tp$ and $\alpha \in TVar$,

$$\text{m}\mu\text{dfo}(\alpha, \tau) =_{\text{def}} \max \{ n \mid n \in \omega,\, \text{(A.32) holds} \} \tag{A.31}$$

be the *maximum μ-depth of a free occurrence of* α in τ, where

$$\left.\begin{aligned} &(\exists\, p_1, \ldots, p_n,\, p \in Pos(\tau)) \\ &\qquad \big[\, p_1 < p_2 < \ldots < p_n < p \;\;\&\;\; \tau|_p = \alpha \;\;\& \\ &\qquad\quad \&\;\; \mu Pos(\tau) \cap Pref(p) = \{p_1, \ldots, p_n, p\} \;\;\& \\ &\qquad\quad \&\;\; (\forall i \in \{1, \ldots, n\})\, (\forall \rho \in \mu Tp)\, [\, \tau|_p \neq \mu\alpha.\, \rho \,] \,\big] \,. \end{aligned}\right\} \tag{A.32}$$

And thirdly, we define, for all $\tau \in \mu Tp$, by

$$\text{m}\mu\text{bd}(\tau) =_{\text{def}} \left\{\begin{aligned} &0 \qquad\qquad \ldots\ldots \text{ if } \tau \text{ has no subterm of the form } \mu\alpha.\, \rho \\ &1 + \max \{ \text{m}\mu\text{dfo}(\alpha, \rho) \mid \alpha \in TVar,\, \rho \in \mu Tp,\, \mu\alpha.\, \rho \trianglelefteq \tau \} \\ &\qquad\qquad\quad \ldots\ldots \text{ else} \end{aligned}\right\} \tag{A.33}$$

the *maximal μ-binding depth* of τ.

Before we show the statement of the lemma, we gather some obvious properties of the three notions defined above. For all $\tau \in \mu Tp$ and $\alpha \in \text{fv}(\tau)$, the following four statements holds:

$$\text{nl}\mu\text{b}(\tau) \leq \text{mns}\mu\text{b}(\tau) \,, \tag{A.34}$$

$$\text{mns}\mu\text{b}(\tau) \leq |\tau| \,, \tag{A.35}$$

$$\text{m}\mu\text{bd}(\tau) \leq |\tau| \,, \tag{A.36}$$

$$\alpha \in \text{fv}(\tau) \implies \text{nl}\mu\text{b}(\mu\alpha.\, \tau) \leq \text{m}\mu\text{dfo}(\alpha, \tau) + 1 \leq \text{m}\mu\text{bd}(\mu\alpha.\, \tau) \,. \tag{A.37}$$

$$\tau \trianglelefteq \sigma \implies \text{mns}\mu\text{b}(\tau) \leq \text{mns}\mu\text{b}(\sigma) \;\;\&\;\; \text{m}\mu\text{bd}(\tau) \leq \text{m}\mu\text{bd}(\sigma) \tag{A.38}$$

All of these four assertions are easy to verify.

We will prove the statement of the lemma by showing

$$(\forall \tau, \sigma \in \mu Tp)\left[\, \sigma \twoheadrightarrow_{\mathrm{roud}\bot} \tau \;\;\Longrightarrow\;\; \mathrm{mns}\mu\mathrm{b}(\tau) \le 2|\sigma| \,\right], \tag{A.39}$$

which is slightly stronger and which implies the lemma because of the assertion connected with (A.34).

We start by observing that the 'measure' $\mathrm{mns}\mu\mathrm{b}(\cdot)$ is non-increasing during $\twoheadrightarrow_{\mathrm{roud}\bot}$-reduction sequences: that is, for all $\sigma, \tau \in \mu Tp$,

$$\sigma \twoheadrightarrow_{\mathrm{roud}\bot} \tau \quad\Longrightarrow\quad \mathrm{m}\mu\mathrm{bd}(\tau) \le \mathrm{m}\mu\mathrm{bd}(\sigma) \tag{A.40}$$

holds. This can be shown by induction on the length of $\rightarrow_{\mathrm{roud}\bot}$-reduction sequences using the statements that, for all $\tau_1, \tau_2 \in \mu Tp$,

$$\begin{aligned}(\,\tau_1 \rightarrow_{\mathrm{ren}} \tau_2\,) \;\vee\; (\,\tau_1 \rightarrow_{\text{out-dec}} \tau_2\,) \;\vee\; (\,\tau_1 \rightarrow_{\text{out-}(\mu\text{–}\bot)'} \tau_2\,) \quad\Longrightarrow& \\ \Longrightarrow\quad \mathrm{m}\mu\mathrm{bd}(\tau_2) \le \mathrm{m}\mu\mathrm{bd}(\tau_1)&\end{aligned} \tag{A.41}$$

holds, which can verified in a straightforward way, and that, for all $\tau_1, \tau_2 \in \mu Tp$,

$$\tau_1 \rightarrow_{\text{out-unf}} \tau_2 \quad\Longrightarrow\quad \mathrm{m}\mu\mathrm{bd}(\tau_2) \le \mathrm{m}\mu\mathrm{bd}(\tau_1) \tag{A.42}$$

holds. The latter statement can be restated as the assertion that for all $\alpha \in TVar$ and $\rho \in \mu Tp$

$$\mu\alpha.\,\rho \rightarrow_{\text{out-unf}} \rho[\mu\alpha.\,\rho/\alpha] \quad\Longrightarrow\quad \mathrm{m}\mu\mathrm{bd}(\rho[\mu\alpha.\,\rho/\alpha]) \le \mathrm{m}\mu\mathrm{bd}(\mu\alpha.\,\rho) \tag{A.43}$$

holds. This statement, however, can be justified, in a slightly informal way here, as follows: if a reduction step $\mu\alpha.\,\rho \rightarrow_{\text{out-unf}} \rho[\mu\alpha.\,\rho/\alpha]$ is assumed, for some $\alpha \in TVar$ and $\rho \in TVar$, then due to the admissibility of the substitution expression $\rho[\mu\alpha.\,\rho/\alpha]$ it cannot be the case that some μ-binding $\mu\beta$ in ρ at a position in $Pos(\rho)$ 'catches' an additional bound occurrence of β in $\rho[\mu\alpha.\,\rho/\alpha]$ at a position in $Pos(\rho[\mu\alpha.\,\rho/\alpha])\backslash \backslash Pos(\rho)$. Therefore the maximal μ-binding depth does not increase in reduction steps of $\rightarrow_{\text{out-unf}}$.

Furthermore we need the following statement: for all $\tau_1, \tau_2 \in \mu Tp$,

$$\tau_1 \rightarrow_{\text{out-unf}} \tau_2 \;\;\Longrightarrow\;\; \mathrm{mns}\mu\mathrm{b}(\tau_2) \le \max\{\mathrm{mns}\mu\mathrm{b}(\tau_1),\, 2.\mathrm{m}\mu\mathrm{bd}(\tau_1)\} \tag{A.44}$$

holds. If $\tau_1 \equiv \mu\alpha.\,\rho$ for some $\rho \in \mu Tp$ and $\alpha \in TVar$ such that $\alpha \notin \mathrm{fv}(\tau)$, then the bound on the right-hand side of (A.44) follows, since in this case $\tau_2 \equiv \rho$ holds, as a consequence of (A.38). In the other case, where $\tau_1 \equiv \mu\alpha.\,\rho$ for some $\rho \in \mu Tp$ and $\alpha \in TVar$ such that $\alpha \in \mathrm{fv}(\rho)$, and where $\tau_1 \rightarrow_{\text{out-unf}} \tau_2$ holds for some $\tau_2 \in \mu Tp$, the bound in (A.44) can be justified in the following way:

$$\begin{aligned}\mathrm{mns}\mu\mathrm{b}(\tau_2) &= \mathrm{mns}\mu\mathrm{b}(\rho[\mu\alpha.\,\rho/\alpha]) \\ &\le \max\Big\{\, \mathrm{mns}\mu\mathrm{b}(\mu\alpha.\,\rho),\;\; \underbrace{\mathrm{m}\mu\mathrm{dfo}(\alpha,\rho)}_{\le\, \mathrm{m}\mu\mathrm{bd}(\mu\alpha.\,\rho)-1} \;+\; \underbrace{\mathrm{nl}\mu\mathrm{b}(\mu\alpha.\,\tau)}_{\underset{(\mathrm{A.34})}{\le}\; \mathrm{m}\mu\mathrm{bd}(\mu\alpha.\,\tau)} \;\Big\} \\ &\le \max\big\{\, \mathrm{mns}\mu\mathrm{b}(\mu\alpha.\,\rho),\, 2.\mathrm{m}\mu\mathrm{bd}(\mu\alpha.\,\rho)\,\big\} \\ &= \max\big\{\, \mathrm{mns}\mu\mathrm{b}(\tau_1),\, 2.\mathrm{m}\mu\mathrm{bd}(\tau_1)\,\big\}\,.\end{aligned}$$

Together with the easily verifiable statement that

$$\begin{aligned}(\,\tau_1 \to_{\text{ren}} \tau_2\,) \;\vee\; (\,\tau_1 \to_{\text{out-dec}} \tau_2\,) \;\vee\; (\,\tau_1 \to_{\text{out-}(\mu\text{–}\bot)'} \tau_2\,) \quad \Longrightarrow \\ \Longrightarrow \quad \text{mns}\mu\text{b}(\tau_2) \le \text{mns}\mu\text{b}(\tau_1) \end{aligned} \tag{A.45}$$

holds for all $\tau_1, \tau_2 \in \mu Tp$, the statement connected with (A.44) can now be used to show

$$\sigma \twoheadrightarrow_{\text{roud}\bot} \tau \quad \Longrightarrow \quad \text{mns}\mu\text{b}(\tau) \le \max\big\{\text{mns}\mu\text{b}(\sigma),\, 2|\sigma|\big\}\ , \tag{A.46}$$

for all $\sigma, \tau \in \mu Tp$, by induction on the length of a $\to_{\text{roud}\bot}$-reduction sequence from σ to τ. In the induction step concerning a reduction step of $\to_{\text{out-unf}}$ at the end of a $\to_{\text{roud}\bot}$-reductions sequence, where we have

$$\sigma \;\twoheadrightarrow_{\text{roud}\bot}\; \tau_1 \;\to_{\text{out-unf}}\; \tau_2$$

for some $\sigma, \tau_1, \tau_2 \in \mu Tp$, we can now reason as follows:

$$\begin{aligned} \text{mns}\mu\text{b}(\tau_2) \;&\underset{(\text{A.44})}{\le}\; \max\Big\{\,\text{mns}\mu\text{b}(\tau_1),\, 2.\; \underbrace{\text{m}\mu\text{bd}(\tau_1)}_{\underset{(\text{A.40})}{\le}\ \text{m}\mu\text{bd}(\sigma)\ \underset{(\text{A.35})}{\le}\ |\sigma|}\,\Big\} \\ &\underset{(\text{I.H.})}{\le}\; \max\{\text{mns}\mu\text{b}(\sigma), 2.|\sigma|\}\ . \end{aligned}$$

Now the strengthening (A.39) of the assertion of the lemma follows from the statement connected with (A.46) by using the mentioned fact that (A.35) holds for all $\tau \in \mu Tp$. In this way we have shown the lemma.

□

Appendix B

Abstract Proof Systems

The purpose of this appendix is to formally underpin the results concerning derivability and admissibility of inference rules that are reported in Chapter 4. For this purpose, we introduce the notions "abstract pure Hilbert system" (APHS) and "abstract natural deduction system" (ANDS) in Section B.1 and Section B.2 below, respectively. We respectively adapt the notions of derivability, admissibility, and correctness of inference rules to APHS's and ANDS's, and subsequently gather basic facts about these notions and their relationship towards each other. Eventually the focus of our interest is, in each kind of proof systems, on statements that clarify the relationship between derivability or admissibility of a rule R in a system $\mathcal{S}$ and the possibility to eliminate applications of R from derivations in extensions of $\mathcal{S}$ by adding R as a new rule.

The concept of APHS defined in Section B.1 is a slight reformulation of the concept of n-AHS ("abstract Hilbert system with names for axioms and rules") that we have introduced and investigated earlier in [Gra03a] (see also the short overview [Gra04b]). In Section B.1 only a small part of the results developed in the mentioned report is treated, with the statements presented here being generally straightforward adaptations to APHS's of statements given there. The concept of ANDS and the investigation of rule derivability and admissibility in ANDS's, the topics treated in Section B.2, are a more recent extension to natural-deduction systems of the study, in the report referred to above, of rule derivability and admissibility in "abstract Hilbert systems".

Only a very small fraction of the statements given here will be proved. As a consequence, this appendix has to be read as of a brief outline of results that we have obtained for concepts that we define here precisely. In the case of APHS's we may however refer to the mentioned report [Gra03a]: there, for practically all of the results given here, closely corresponding statements[1] can be found, together with detailed proofs.

[1] We mean statements that correspond to statements given here within the concepts developed there of "abstract Hilbert system" (AHS) and of "abstract Hilbert system with names" (n-AHS).

B.1 Abstract Pure Hilbert Systems

For the purpose of collecting general results about the notions or derivability and admissibility of inference rules in pure Hilbert systems, we introduce the framework of "abstract pure Hilbert system". In such systems, which are defined by analogy with abstract rewrite systems, it is abstracted away from the syntax of the formula language in a pure Hilbert system, and consequently also from the specific ways how rules can be defined syntactically. In an abstract pure Hilbert system a rule is a set of instances that is endowed with a premise and a conclusion function, which respectively assign a finite sequence of premises and a conclusion to every instance. After defining the notion of APHS-rule with respect to a given set (of formulas), abstract pure Hilbert systems are then defined as consisting of a set of formulas, a set of named axioms (which are named formulas), and a set of named rules that 'operate' on the set of formulas of the system.

The notion of "abstract pure Hilbert system" is analogous to, and in fact has been motivated by, the notion of "abstract rewriting system" in the formulation of van Oostrom and de Vrijer in [vOdV02] and in [Ter03, p. 317]. There, an ARS $\mathcal{A}$ is defined as a quadruple $\langle A, \Phi, \mathsf{src}, \mathsf{tgt}\rangle$ in which A and Φ are sets whose members are respectively called *objects* and *steps*, and $\mathsf{src}, \mathsf{tgt} : \Phi \to A$ are the *source* and the *target functions* of $\mathcal{A}$.

We start by defining the abstract notion of rule that underlies the notion of "abstract pure Hilbert system", which will be defined subsequently.

Definition B.1.1. (An abstract notion of rule). Let Fo be a set. An *APHS-rule* (*a rule for an "abstract pure Hilbert system"*)[2] on (formulas of) Fo is a triple of the form $\langle \mathit{Insts}, \mathsf{prem}, \mathsf{concl}\rangle$, where

- Insts is a set, the members of which are called the *instances* or *applications of* R,
- and $\mathsf{prem} : \mathit{Insts} \to \mathit{Seqcs}_{\mathrm{f}}(Fo)$ and $\mathsf{concl} : \mathit{Insts} \to Fo$ are the *premise* and *conclusion* functions of R.

We will use the symbolic denotations Insts_R, prem_R and concl_R, whenever we want to refer directly to the instance set, the premise and conclusion functions of a rule R, respectively. And we will use ι as a syntactical variable for instances.

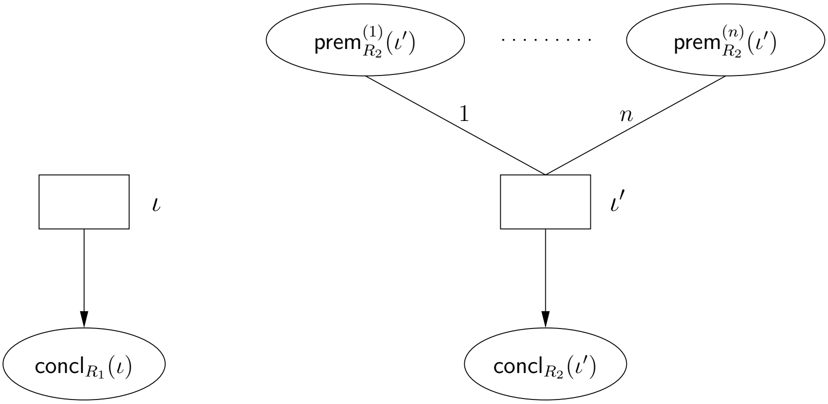
For all sets Fo, we furthermore denote by $\mathfrak{R}(Fo)$ the class of APHS-rules on Fo. ⊠

In addition to the functions prem and concl associated with a rule, we define for auxiliary purposes the functions arity and $\mathsf{prem}^{(i)}$: for every set Fo of formulas and for every rule $R = \langle \mathit{Insts}, \mathsf{prem}, \mathsf{concl}\rangle$ on Fo, we introduce

$$\text{the function} \quad \mathsf{arity} : \mathit{Insts} \to \omega$$
$$\iota \mapsto \mathsf{arity}(\iota) =_{\mathrm{def}} \lg(\mathsf{prem}(\iota)) \ ,$$

[2] The concept of "abstract pure Hilbert system" is defined below in Definition B.1.2.

Figure B.1: Visualization as 'hypergraph hyperedges' of two kinds of applications of APHS-rules: of a zero premise application ι of a rule R_1 and of an application ι' of a rule R_2 such that ι' has $n = \mathsf{arity}(\iota') \in \omega\backslash\{0\}$ premises.


$\mathsf{prem}^{(1)}_{R_2}(\iota')$
....
$\mathsf{prem}^{(n)}_{R_2}(\iota')$
1
n
ι
ι'
$\mathsf{concl}_{R_1}(\iota)$
$\mathsf{concl}_{R_2}(\iota')$

and, for all $i \in \omega\backslash\{0\}$,

$$\text{the partial functions} \quad \mathsf{prem}^{(i)} : \mathit{Insts} \rightharpoonup \mathit{Fo}$$
$$\iota \mapsto \mathsf{prem}^{(i)}(\iota) =_{\text{def}} \text{proj}_i(\mathsf{prem}(\iota)) \ .$$

arity assigns to every instance ι of R the number of its premises. And for all $i \in \omega\backslash\{0\}$, $\mathsf{prem}^{(i)}(\iota)$ assigns to every instance ι of R its i-th premise, whenever this exists; otherwise $\mathsf{prem}^{(i)}(\iota)$ is undefined. We will use the denotations arity_R and $\mathsf{prem}^{(i)}_R$, whenever we want to make the dependence of arity and $\mathsf{prem}^{(i)}$ upon R explicit. Using these definitions, a visualization as 'hyperedges' of a 'hypergraph' (see [Plu93, p.12]) of two kinds of applications of APHS-rules (applications without premises and with a finite, nonzero number of premises) is given in Figure B.1.

Now we introduce "abstract pure Hilbert systems" as structures consisting of a set of formulas, a set of names, a set of named axioms, and a set of named rules.

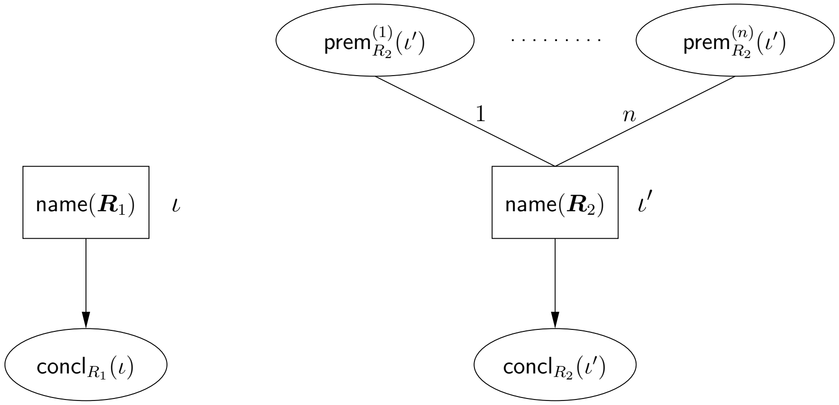
Definition B.1.2. (Abstract pure Hilbert Systems). An *abstract pure Hilbert system* (an APHS) $\mathcal{H}$ is a quadruple $\langle \mathit{Fo}, \mathit{Na}, n\mathcal{A}x, n\mathcal{R} \rangle$ where

- Fo, Na, $n\mathcal{A}x$ and $n\mathcal{R}$ are sets whose elements are respectively called the *formulas of* $\mathcal{H}$, the *names* (for axioms and rules) *in* $\mathcal{H}$, the *named axioms of* $\mathcal{H}$, and the *named rules of* $\mathcal{H}$; we demand $\mathit{Fo} \neq \emptyset$, i.e. that the formula set be nonempty;

- $n\mathcal{A}x \subseteq \mathit{Fo} \times \mathit{Na}$, i.e. the named axioms of $\mathcal{H}$ are tuples with formulas of $\mathcal{H}$ as first and names in $\mathcal{H}$ as second components;

- $n\mathcal{R} \subseteq \mathfrak{R}(\mathit{Fo}) \times \mathit{Na}$, i.e. the named rules of $\mathcal{H}$ are tuples that have APHS-rules on Fo in their first and names in $\mathcal{H}$ in their second components;

Figure B.2: Visualization as labeled hyperedges (in a concept of 'hypergraph') of two kinds of rule applications in an APHS $\mathcal{H}$ with set $n\mathcal{R}$ of named rules: illustration of a zero premise application ι of a named rule $\boldsymbol{R}_1 = \langle R_1, \mathsf{name}(\boldsymbol{R}_1)\rangle \in n\mathcal{R}$, and of an application ι' of a named rule $\boldsymbol{R}_2 = \langle R_2, \mathsf{name}(\boldsymbol{R}_2)\rangle \in n\mathcal{R}$ such that ι' has $n = \mathsf{arity}(\iota') \in \omega\backslash\{0\}$ premises.


$\mathsf{prem}^{(1)}_{R_2}(\iota')$
....
$\mathsf{prem}^{(n)}_{R_2}(\iota')$
1
n
$\mathsf{name}(\boldsymbol{R}_1)$
ι
$\mathsf{name}(\boldsymbol{R}_2)$
ι'
$\mathsf{concl}_{R_1}(\iota)$
$\mathsf{concl}_{R_2}(\iota')$

(we will use the boldface symbol $\boldsymbol{R}$, possibly with subscripts, superscripts or accents attached to it, as syntactical variable for named rules; for a named rule $\boldsymbol{R}$ we denote by $\mathsf{rule}(\boldsymbol{R})$, and by $\mathsf{name}(\boldsymbol{R})$, the first, and respectively the second component of $\boldsymbol{R}$ (the *rule* and the *name* of $\boldsymbol{R}$); furthermore we will frequently write, for example, $\boldsymbol{R} = \langle R, \mathsf{name}(\boldsymbol{R})\rangle$ and use hereby (and in other situations) the non-boldface symbol R for the rule component $\mathsf{rule}(\boldsymbol{R})$ of a named rule $\boldsymbol{R}$);

- for the named axioms and the named rules of $\mathcal{H}$ furthermore the following holds:

$$(\forall \langle A, name\rangle \in n\mathcal{A}x)\,(\forall \boldsymbol{R} \in n\mathcal{R})\,\big[\,name \neq \mathsf{name}(\boldsymbol{R})\,\big]\ ,$$

i.e. names of named axioms are different from names of named rules, and

$$(\forall \boldsymbol{R}_1, \boldsymbol{R}_2 \in n\mathcal{R})\,\big[\,\boldsymbol{R}_1 \neq \boldsymbol{R}_2 \implies \mathsf{name}(\boldsymbol{R}_1) \neq \mathsf{name}(\boldsymbol{R}_2)\,\big]\ .$$

i.e. different rules are differently named in $n\mathcal{R}$ (but it is not excluded that the same rule may occur with different names in $n\mathcal{R}$).

We denote by $\mathfrak{H}$ the class of all abstract pure Hilbert systems. If, for a some abstract pure Hilbert system $\mathcal{H}$, we want to refer to its set of formulas, its set of names, its set of named axioms, or its set of named rules, then we will use the symbolic denotations $Fo_{\mathcal{H}}$, $Na_{\mathcal{H}}$ $n\mathcal{A}x_{\mathcal{H}}$ or $n\mathcal{R}_{\mathcal{H}}$, respectively.

⊠

A visualization of labeled hyperedges of a hypergraph (see [Plu93, p.12]) of two kinds of rule applications in an APHS (applications without premises, and with a finite nonzero number of premises) is given in Figure B.2. This illustration of rule applications is oriented at the way[3] how labeled hyperedges of hypergraphs are drawn in [Plu93].

Now we introduce derivations in an APHS $\mathcal{H}$ as *prooftrees* in the sense of [TS00], and explained in Chapter 2: as trees in which the nodes are labeled by formulas and in which the edges make part of rule applications and are not drawn, but are replaced by horizontal lines that represent applications. Axioms and assumptions appear as top nodes and lower nodes are formed by applications of rules. Furthermore, occurrences of axioms and of inferences that correspond to rule applications will furthermore be labeled by the names of the respective axioms or rules. Still, the usual notation for prooftrees will also be applied for derivations in APHS's.

Definition B.1.3. (Derivations in abstract pure Hilbert systems). Let an APHS $\mathcal{H} = \langle Fo, Na, n\mathcal{A}x, n\mathcal{R}\rangle$ be given.

The set $\mathcal{D}er(\mathcal{H})$ of *derivations in* $\mathcal{H}$ is defined as the smallest set that is closed under the three generation steps (i), (ii), and (iii) which are detailed below. Simultaneously with this definition also the three functions

$$\mathsf{assm}\colon \mathcal{D}er(\mathcal{H}) \to \mathcal{M}_{\mathrm{f}}(Fo) \qquad \mathsf{concl}\colon \mathcal{D}er(\mathcal{H}) \to Fo \qquad |\cdot|\colon \mathcal{D}er(\mathcal{H}) \to \omega$$

are defined that assign to a derivation $\mathcal{D}$ in $\mathcal{H}$ the multiset $\mathsf{assm}(\mathcal{D})$ of *assumptions* of $\mathcal{D}$, the *conclusion* $\mathsf{concl}(\mathcal{D})$ of $\mathcal{D}$, and the *(rule application) depth* $|\mathcal{D}|$ of $\mathcal{D}$, respectively.

(i) For every named axiom $\langle A, name\rangle \in n\mathcal{A}x$, the prooftree $\mathcal{D}$ of the form

$$\frac{(name)}{A}$$

is a derivation in $\mathcal{H}$ with conclusion A, without assumptions, and with depth zero, i.e. $\mathcal{D} \in \mathcal{D}er(\mathcal{H})$ holds, and it is stipulated here that $\mathsf{concl}(\mathcal{D}) =_{\mathrm{def}} A$, $\mathsf{assm}(\mathcal{D}) =_{\mathrm{def}} \mathrm{mset}(\emptyset)$, and $|\mathcal{D}| =_{\mathrm{def}} 0$.

(ii) For all formulas $A \in Fo$, the prooftree $\mathcal{D}$ consisting only of the assumption

$$A$$

is a derivation in $\mathcal{H}$ with conclusion A, with precisely one assumption (the formula A), and with depth zero, that is, $\mathcal{D} \in \mathcal{D}er(\mathcal{H})$ holds, and the formal stipulations $\mathsf{concl}(\mathcal{D}) =_{\mathrm{def}} A$, $\mathsf{assm}(\mathcal{D}) =_{\mathrm{def}} \mathrm{mset}(\{A\})$, and $|\mathcal{D}| =_{\mathrm{def}} 0$ are agreed.

[3] This is described in [Plu93] on p. 12 as follows: "In pictures of hypergraphs, nodes are drawn as circles or ovals, and hyperedges as boxes, in both cases with inscribed labels. Lines connect a hyperedge with its source nodes, while arrows point to the target nodes. Lines and arrows are numbered corresponding to the order of nodes in source and target strings." – In our illustration of rule applications in an APHS, however, we have labeled the nodes (formulas) by themselves (if two or more premises of a rule application coincide, then the respective source nodes coincide as well).

(iii) Let $\boldsymbol{R} \in n\mathcal{R}$ (i.e. let $\boldsymbol{R}$ be a named rule of $\mathcal{H}$), let its rule component $\mathsf{rule}(\boldsymbol{R})$ be designated by $R = \langle \mathit{Insts}_R, \mathsf{prem}_R, \mathsf{concl}_R \rangle$, and let ι be an instance of R (i.e. let $\iota \in \mathit{Insts}_R$). Depending on the arity of ι, we distinguish two cases:

Case 1. $\mathsf{arity}_R(\iota) = 0$.
Given that $\mathsf{concl}_R(\iota) = A$, where A is a formula of $\mathcal{H}$, the prooftree

$$\frac{}{A}\ \mathsf{name}(R)$$

is a derivation $\mathcal{D}$ in $\mathcal{H}$ with conclusion A, without assumptions, and with depth zero, i.e. here $\mathcal{D} \in \mathcal{D}er(\mathcal{H})$ holds, and the settings $\mathsf{concl}(\mathcal{D}) =_{\text{def}} A$, $\mathsf{assm}(\mathcal{D}) =_{\text{def}} \text{mset}(\emptyset)$, and $|\mathcal{D}| =_{\text{def}} 0$ are stipulated.

Case 2. $\mathsf{arity}_R(\iota) = n \in \omega \backslash \{0\}$.
Assuming that $\mathsf{prem}_R(\iota) = \langle A_1, \ldots, A_n \rangle$ and that $\mathsf{concl}_R(\iota) = A$, for some formulas $A, A_1, \ldots, A_n$ of $\mathcal{H}$, and assuming further that $\mathcal{D}_1, \ldots, \mathcal{D}_n$ are derivations in $\mathcal{H}$ with respective conclusions $A_1, \ldots, A_n$ (in particular this means that $\mathsf{concl}(\mathcal{D}_i) = A_i$ holds for all $i \in \{1, \ldots, n\}$), the prooftree of the form

$$\frac{\begin{array}{ccc} \mathcal{D}_1 & & \mathcal{D}_n \\ A_1 & \ldots & A_n \end{array}}{A}\ \mathsf{name}(R)$$

is a derivation $\mathcal{D}$ in $\mathcal{H}$ with conclusion $\mathsf{concl}(\mathcal{D}) =_{\text{def}} A$ and with its multiset of assumptions and its depth defined according to

$$\mathsf{assm}(\mathcal{D}) =_{\text{def}} \biguplus_{i=1}^{n} \mathsf{assm}(\mathcal{D}_i) \,,$$

$$|\mathcal{D}| =_{\text{def}} 1 + \max\{\, |\mathcal{D}_i| \mid i \in \omega,\ 1 \le i \le n \,\} \,.$$

Furthermore we define by

$$\mathcal{D}er_{\emptyset}(\mathcal{H}) =_{\text{def}} \{\mathcal{D} \in \mathcal{D}er(\mathcal{H}) \mid \text{set}(\mathsf{assm}(\mathcal{D})) = \emptyset\}$$

the set of all derivations in $\mathcal{H}$ without assumptions. ⊠

Having defined a precise notion of derivation in APHS's, together with assumption and conclusion functions on the set of derivations, we can now introduce a 'usual', or 'standard', notion of consequence relation on an arbitrary APHS $\mathcal{H}$, which can be used to succinctly express propositions such as "a formula A is derivable in $\mathcal{H}$ from assumptions contained in the set Σ of formulas (thereby being allowed to use each of these assumptions an arbitrary number of times)."

Definition B.1.4 (The 'standard' consequence relation on an APHS). Let $\mathcal{H}$ be an APHS with set *Fo* of formulas. We define the consequence relation $\vdash_{\mathcal{H}}$, where $\vdash_{\mathcal{H}} \subseteq \mathcal{P}(\mathit{Fo}) \times \mathit{Fo}$, by stipulating for all $A \in \mathit{Fo}$ and *sets* $\Sigma \in \mathcal{P}(\mathit{Fo})$

$$\Sigma \vdash_{\mathcal{H}} A \quad \Longleftrightarrow_{\text{def}} \quad (\exists \mathcal{D} \in \mathcal{D}er(\mathcal{H}))\big[\, \text{set}(\mathsf{assm}(\mathcal{D})) \subseteq \Sigma \ \&\ \mathsf{concl}(\mathcal{D}) = A \,\big] \,;$$

if, for some $A \in Fo$ and $\Sigma \in \mathcal{P}(Fo)$, $\Sigma \vdash_{\mathcal{H}} A$ holds, then we say that A *can be derived in* $\mathcal{H}$ *from assumptions in* Σ. ⊠

Quite obviously, other 'natural' consequence relations that correspond to different forms of 'resource-consciousness' when considering derivations could be defined similarly as well: for instance, a consequence relation $\vdash^{(\mathrm{m})} \subseteq \mathcal{M}_{\mathrm{f}}(Fo_{\mathcal{H}}) \times Fo_{\mathcal{H}}$ with respect to which a statement $\Gamma \vdash^{(\mathrm{m})}_{\mathcal{H}} A$ holds, for arbitrary $\Gamma \in \mathcal{M}_{\mathrm{f}}(Fo_{\mathcal{H}})$ and $A \in Fo_{\mathcal{H}}$, if and only if there exists a derivation $\mathcal{D}$ in $\mathcal{H}$ with Γ as its multiset of assumptions and with conclusion A. We do not introduce this notion here, but refer to [Gra03a] instead, where $\vdash^{(\mathrm{m})}$ is introduced together with the usual consequence relation $\vdash$ and a further one $\vdash^{(\mathrm{s})}$, and where also alternative notions of derivability induced by the consequence relations $\vdash^{(\mathrm{s})}$ and $\vdash^{(\mathrm{m})}$ are studied.

Based on the introduced notion of consequence relation in an APHS, we now define the notions of "theorem" in an APHS, of the "theory" of an APHS, and of two APHS's being "equivalent" in the obvious ways.

Definition B.1.5. (Theorems, theory of an APHS, equivalent APHS's). Let $\mathcal{H}$ be an APHS. A formula $A \in Fo_{\mathcal{H}}$ is a *theorem* of $\mathcal{H}$ if and only if $\emptyset \vdash_{\mathcal{H}} A$, i.e. iff there exists a derivation $\mathcal{D}$ in $\mathcal{H}$ from the empty set of assumptions and with conclusion A; in this case we write $\vdash_{\mathcal{H}} A$ for $\emptyset \vdash_{\mathcal{H}} A$. The *theory of* $\mathcal{H}$ is the set $Th(\mathcal{H}) = \{A \in Fo_{\mathcal{H}} \mid \vdash_{\mathcal{H}} A\}$ of theorems of $\mathcal{H}$.

Let $\mathcal{H}_1$ and $\mathcal{H}_2$ be APHS's. We say that $\mathcal{H}_1$ and $\mathcal{H}_2$ are *equivalent* (which we denote by $\mathcal{H}_1 \sim_{th} \mathcal{H}_2$) if and only if $Th(\mathcal{H}_1) = Th(\mathcal{H}_2)$ holds. ⊠

Extensions of systems of formal logic are often defined in the following way (see, for example, the special case of "first-order theories" treated in [Shoe67, p. 41]): A formal system $\mathcal{S}'$ is called an "extension" of a formal system $\mathcal{S}$ if the language of $\mathcal{S}'$ is an extension of the language of $\mathcal{S}$ and if every theorem of $\mathcal{S}$ is also a theorem of $\mathcal{S}'$. Furthermore, an extensions of a theory $\mathcal{S}$ that arises from $\mathcal{S}$ in the special way of adding new function and/or predicate symbols, together with defining axioms, to the non-logical symbols and respectively to the axioms of $\mathcal{S}$ is called an "extension by definitions" of $\mathcal{S}$ in [Shoe67]. In the following definition we introduce a similar term, and some useful notation, for the following kind of extensions of APHS's which arise naturally in the study of rule admissibility: extensions of an APHS that arise by adding new formulas, new axioms, and/or new rules to the sets of formulas, the set of axioms, and to the set of rules of the system, respectively.

Definition B.1.6 (Extensions by enlargement of APHS's). For the stipulations in the following three items, let $\mathcal{H} = \langle Fo, Na, n\mathcal{A}x, n\mathcal{R}\rangle$ be an APHS.

(i) An *extension by enlargement of* $\mathcal{H}$ is an APHS $\mathcal{H}' = \langle Fo', Na', n\mathcal{A}x', n\mathcal{R}'\rangle$ that results from $\mathcal{H}$ by extending the sets of formulas, of names, of named axioms and/or of named rules of $\mathcal{H}$, i.e. for which the following holds:

$$Fo \subseteq Fo' \;\;\&\;\; Na \subseteq Na' \;\;\&\;\; n\mathcal{A}x \subseteq n\mathcal{A}x' \;\;\&\;\; n\mathcal{R} \subseteq n\mathcal{R}' \;.$$

(ii) Let R be an APHS-rule on *Fo*. An *extension of $\mathcal{H}$ by adding R as a new rule* (or shorter, an *extension of $\mathcal{H}$ by adding the new rule R*) is an extension by enlargement of the form

$$\mathcal{H}' = \langle Fo, Na', n\mathcal{A}x, n\mathcal{R} \cup \{\langle R, name\rangle\}\rangle$$

for some $name \in Na'$ such that $\langle R, name\rangle \notin n\mathcal{R}$ holds. An extension of $\mathcal{H}$ by adding R as a new rule will generally be denoted by $\mathcal{H}^{(+R)}$.

(iii) Let $\Sigma \subseteq Fo$ be arbitrary. An *extension of $\mathcal{H}$ by adding the formulas of Σ as new axioms* (or shorter, an *extension of $\mathcal{H}$ by adding the new axioms Σ*) is an extension by enlargement of the form

$$\mathcal{H}' = \langle Fo, Na', n\mathcal{A}x \cup \tilde{\Sigma}, n\mathcal{R}\rangle$$

for some set $\tilde{\Sigma} \subseteq \Sigma \times Na'$ such that

$$\mathrm{proj}_1(\tilde{\Sigma}) = \Sigma' \quad \& \quad \mathrm{proj}_2(\tilde{\Sigma}) \cap \mathrm{proj}_2(n\mathcal{A}x) = \emptyset$$

holds (this means that the set of axioms of $\mathcal{H}$ is extended, for all $A \in \Sigma$, by a pair $\langle A, name\rangle$ where $name \in Na'$ is a name in $\mathcal{H}'$ that does not occur as a name in an named axiom[4] of $\mathcal{H}$). An extension of an APHS $\mathcal{H}$ by adding the new axioms Σ will generally be denoted by the expression $\mathcal{H}^{(+\Sigma)}$.

⊠

The following proposition contains the assertion that in an extension $\mathcal{S}^{(+R)}$ of an APHS $\mathcal{S}$ by adding a new rule R the specific name chosen for the named version of R added in $\mathcal{S}^{(+R)}$ does not matter for the consequence relation $\vdash_{\mathcal{S}^{(+R)}}$ on $\mathcal{S}^{(+R)}$; and a similar assertion holds for extensions of an APHS by adding new axioms.

Proposition B.1.7. *Let $\mathcal{H}$ be an APHS, let R be an APHS-rule on $Fo_{\mathcal{H}}$, and let $\Sigma \subseteq Fo_{\mathcal{H}}$. Then the following two statements hold:*

$$\big(\forall\, \mathcal{H}_1, \mathcal{H}_2,\ \textit{extensions of } \mathcal{H} \textit{ by adding the new rule } R\big)\ \big[\,(\vdash_{\mathcal{H}_1} = \vdash_{\mathcal{H}_2})\ \&\ (\mathcal{H}_1 \sim_{th} \mathcal{H}_2)\,\big]\,, \tag{B.1}$$

$$\big(\forall\, \mathcal{H}_1, \mathcal{H}_2,\ \textit{extensions of } \mathcal{H} \textit{ by adding the new axioms } \Sigma\big)\ \big[\,(\vdash_{\mathcal{H}_1} = \vdash_{\mathcal{H}_2})\ \&\ (\mathcal{H}_1 \sim_{th} \mathcal{H}_2)\,\big]\,. \tag{B.2}$$

That is, all extensions $\mathcal{H}^{(+R)}$ of $\mathcal{H}$ by adding R as a new rule have the same consequence relation (as defined by Definition B.1.4) and they are equivalent, and furthermore, an analogous assertion holds for all extensions by adding new axioms.

Reference to a Proof. For (B.1) we refer to the Sketch of Proof of Proposition B.2.12 where a proof for a similar statement for analogous extensions of “abstract natural-deduction systems” is outlined. And for (B.2) it can be argued similarly.

[4] Due to the definition of APHS’s, *name* cannot be the name component of a named rule of $\mathcal{H}'$, neither.

An easy, later important relationship between the consequence relation $\vdash_{\mathcal{H}}$ in an APHS $\mathcal{H}$ and the consequence relation $\vdash_{\mathcal{H}^{(+\Sigma)}}$ in the extension of $\mathcal{H}$ by adding the new axioms Σ is formulated in the lemma below.

Lemma B.1.8. *Let $\mathcal{H}$ be an APHS with set Fo of formulas. Then for all $A \in Fo$, $\Delta, \Sigma \in \mathcal{P}(Fo)$, and for all extensions $\mathcal{H}^{(+\Sigma)}$ of $\mathcal{H}$ by adding the new axioms Σ, the following holds:*

$$\Delta \vdash_{\mathcal{H}^{(+\Sigma)}} A \quad \Longleftrightarrow \quad \Delta \cup \Sigma \vdash_{\mathcal{H}} A\,. \tag{B.3}$$

Proof. Let $\mathcal{H}$ be an arbitrary APHS with set Fo of formulas. Furthermore, let $A \in Fo$ and $\Sigma, \Delta \in \mathcal{P}(Fo)$ be arbitrary, and let $\mathcal{H}^{(+\Sigma)}$ be an arbitrary extension of $\mathcal{H}$ by adding the new axioms Σ. The equivalence (B.3) is a consequence of the two following observations about a correspondence between derivations $\mathcal{D}$ in $\mathcal{H}^{(+\Sigma)}$ and derivations $\mathcal{D}'$ in $\mathcal{H}$:

- Every derivation $\mathcal{D}$ in $\mathcal{H}^{(+\Sigma)}$ with $\mathsf{concl}(\mathcal{D}) = A$ and $\mathrm{set}(\mathsf{assm}(\mathcal{D})) \subseteq \Delta$ can be modified into a derivation $\mathcal{D}'$ in $\mathcal{H}$ with $\mathsf{concl}(\mathcal{D}') = A$ and $\mathrm{set}(\mathsf{assm}(\mathcal{D}')) \subseteq$ $\subseteq \Delta \cup \Sigma$ by simply replacing each occurrence at the top of the prooftree $\mathcal{D}$ of an axiom C with $C \in \Sigma$ by an occurrence of the assumption C.

- Every derivation $\mathcal{D}'$ in $\mathcal{H}$ with $\mathsf{concl}(\mathcal{D}) = A$ and $\mathrm{set}(\mathsf{assm}(\mathcal{D})) \subseteq \Delta \cup \Sigma$ can be transformed into a derivation $\mathcal{D}$ in $\mathcal{H}^{(+\Sigma)}$ with $\mathsf{concl}(\mathcal{D}) = A$ and with $\mathrm{set}(\mathsf{assm}(\mathcal{D})) \subseteq \Delta$ by simply changing occurrences of assumptions C with $C \in \Sigma$ at the top of $\mathcal{D}'$ into occurrences of axioms C of $\mathcal{H}^{(+\Sigma)}$.

□

Now we are able to adapt the notions of rule admissiblity and derivability to APHS's. More precisely, we define three formally different notions: rule correctness, rule admissibility, and rule derivability in an APHS.

Definition B.1.9 (Rule correctness, admissibility, derivability in APHS's). Let $\mathcal{H}$ be an APHS, and let $R = \langle \mathit{Insts}_R, \mathsf{prem}, \mathsf{concl}\rangle$ be an APHS-rule on $Fo_{\mathcal{H}}$.

(i) The rule R is *correct for $\mathcal{H}$* (R is a *correct rule for $\mathcal{H}$*) if and only if

$$(\forall \iota \in \mathit{Insts}_R)\,\big[\,(\forall A \in \mathrm{set}(\mathsf{prem}(\iota)))\,[\,\vdash_{\mathcal{H}} A\,] \implies \vdash_{\mathcal{H}} \mathsf{concl}(\iota)\,\big] \tag{B.4}$$

holds, i.e. iff the theory of $\mathcal{H}$ is closed under applications of R.

(ii) Let $\mathcal{H}^{(+R)}$ be an arbitrary extension of $\mathcal{H}$ be adding R as a new rule. Then the rule R is *admissible in $\mathcal{H}$* (R is an *admissible rule of $\mathcal{H}$*) if and only if

$$\mathcal{H}^{(+R)} \sim_{th} \mathcal{H} \tag{B.5}$$

holds, i.e. iff extending $\mathcal{H}$ by adding the new rule R with the result $\mathcal{H}^{(+R)}$ does not lead to more theorems in $\mathcal{H}^{(+R)}$.

(iii) The rule R is *derivable in $\mathcal{H}$* (R is a *derivable rule of $\mathcal{H}$*) if and only if

$$(\forall \iota \in \mathit{Insts}_R)\,\big[\,\mathrm{set}(\mathsf{prem}(\iota)) \vdash_{\mathcal{H}} \mathsf{concl}(\iota)\,\big] \tag{B.6}$$

holds, i.e. iff for every application ι of R the conclusion of ι is derivable in $\mathcal{H}$ from the *set* of premises of ι.

And furthermore, we expand these definitions also to named rules in such APHS's whose sets of formulas are contained in $Fo_{\mathcal{H}}$. For this, let $\mathcal{H}'$ be an APHS with $Fo_{\mathcal{H}'} \subseteq Fo_{\mathcal{H}}$, and let $\boldsymbol{R}$ be a named rule of $\mathcal{H}'$. Then $\boldsymbol{R}$ is called *correct for* $\mathcal{H}$, *admissible in* $\mathcal{H}$, or *derivable in* $\mathcal{H}$ if and only if the rule $\mathsf{rule}(\boldsymbol{R})$ of $\boldsymbol{R}$ is correct for $\mathcal{H}$, admissible in $\mathcal{H}$, or derivable in $\mathcal{H}$, respectively.

⊠

The stipulation in Definition B.1.9, (ii) for admissibility of a rule R in an APHS formally depends on the choice of an extension $\mathcal{H}^{(+R)}$ of $\mathcal{H}$ by adding the new rule R; the proposition below asserts that this dependency is inessential.

Proposition B.1.10. *Let $\mathcal{H}$ be an APHS, and let R be an APHS. The stipulation for "R is admissible in $\mathcal{H}$" in (the second sentence of) Proposition B.1.9, (ii) does not depend on the particular choice, made according to the first sentence there, of an extension $\mathcal{H}^{(+R)}$ by adding the new rule R.*

Proof. This is an immediate consequence of statement (B.1) of Proposition B.1.7.

□

Some easy observations about the notions defined in Definition B.1.9 are gathered in the following lemma that is an adaptation to the framework of APHS and a slight reformulation[5] of Lemma 6.14 on p. 70 in the book [HS86] by Hindley and Seldin.

Lemma B.1.11 (Reformulation of a lemma by Hindley and Seldin). *Let $\mathcal{H}$ be an APHS and let R be an APHS-rule on the set of formulas of $\mathcal{H}$. Then the following three statements hold:*

(i) R is correct for $\mathcal{H}$ if and only if R is admissible in $\mathcal{H}$.

(ii) If R is derivable in $\mathcal{H}$, then R is also admissible in $\mathcal{H}$. The implication in the opposite direction does not hold in general.

(iii) If R is derivable in $\mathcal{H}$, then R is derivable in every extension by enlargement of $\mathcal{H}$.

It is a consequence of the statements (ii) and (iii) of Lemma B.1.11 that if a rule R is derivable in an APHS $\mathcal{H}$, then R is admissible in all extensions by enlargement of $\mathcal{H}$. In addition to this fact, the following theorem also states the reverse implication; it thereby provides a characterization of derivability of a rule R in an APHS $\mathcal{H}$ in terms of admissibility of R in extensions by enlargement of $\mathcal{H}$.

Theorem B.1.12. *Let $\mathcal{H}$ be an APHS with Fo as its set of formulas, and let R be an APHS-rule on Fo. Then the following three statements are equivalent:*

[5] The differences concern items (i) and (iii) of Lemma B.1.11. Item (i) is an immediate reformulation of Lemma 6.14, (i), in [HS86] with respect to our definitions of rule admissibility and correctness. The difference between Lemma B.1.9, (iii), and Lemma 6.14, (iii), in [HS86] is the following: Hindley and Seldin do not consider extensions of formal systems that arise by extending the respective set of formulas (however, they consider extensions that result by introducing new axioms and/or new rules).

(i) R is derivable in $\mathcal{H}$.

(ii) R is admissible in all extensions of $\mathcal{H}$ by adding new axioms, i.e. for all sets $\Sigma \subseteq Fo$, R is admissible in every extension $\mathcal{H}^{(+\Sigma)}$ of $\mathcal{H}$ by adding the new axioms Σ.

(iii) R is admissible in every extension by enlargement of $\mathcal{H}$.

Proof. Let $\mathcal{H}$ be an APHS with formula set Fo and let R be an APHS-rule on Fo. We show the equivalence of the statements (i), (ii) and (iii) in the proposition by showing the three implications (i) $\Rightarrow$ (iii), (iii) $\Rightarrow$ (ii), and (ii) $\Rightarrow$ (i). in the three items below, respectively.

(a) The implication (i) $\Rightarrow$ (iii) is an immediate consequence of items (ii) and (iii) in Lemma B.1.11.

(b) The implication (iii) $\Rightarrow$ (ii) is obvious because every AHS of the form $\mathcal{H}^{(+\Sigma)}$, where $\Sigma \in \mathcal{P}(Fo)$, is also an extension by enlargement of $\mathcal{H}$.

(c) To prove (ii) $\Rightarrow$ (i), suppose that R is admissible in all extensions $\mathcal{H}^{(+\Sigma)}$ by adding the new axioms Σ, for some set $\Sigma \in \mathcal{P}(Fo)$. Let ι be an arbitrary application of R, let $\Sigma = \text{set}(\mathsf{prem}(\iota))$, and let $\mathcal{H}^{(+\Sigma)}$ be an arbitrary extension of $\mathcal{H}$ by adding the new axioms Σ. Clearly, $\vdash_{\mathcal{H}^{(+\Sigma)}} A$ holds for all $A \in \text{set}(\mathsf{prem}(\iota))$, and hence, since R is admissible in $\mathcal{H}^{(+\Sigma)}$, it follows that $\vdash_{\mathcal{H}^{(+\Sigma)}} \mathsf{concl}(\iota)$. From this we conclude by Lemma B.1.8 that $\text{set}(\mathsf{prem}(\iota)) \vdash_{\mathcal{H}} \mathsf{concl}(\iota)$. Since ι was an arbitrary application of R in this argument, we have proved that $\text{set}(\mathsf{prem}(\iota)) \vdash_{\mathcal{H}} \mathsf{concl}(\iota)$ holds for all applications of R, which shows (B.6) and hence that R is derivable in $\mathcal{H}$.

□

Theorem B.1.12, whose main assertion (the implications from (ii) and (iii) to (i)) we did not find stated in the literature, is able to provide an explanation for why derivability of a rule R in an APHS $\mathcal{H}$ is a stronger property than admissibility of R in $\mathcal{H}$: derivability of R in $\mathcal{H}$ is preserved under going over to extensions by enlargements of $\mathcal{H}$; “pure admissibility” of R in $\mathcal{H}$ (admissibility of R in $\mathcal{H}$ without R being derivable in $\mathcal{H}$) is not.

For the introduction below of two notions of rule elimination in (derivations of) APHS's, we need a stipulation for the precise circumstances under which a derivation $\mathcal{D}'$ without applications of a certain rule R can be said to ‘demonstrate the same logical assertion’ as another derivation $\mathcal{D}$ containing R-applications; or more abstractly, an agreement about situations in which it is reasonable to say that a derivation $\mathcal{D}'$ is ‘models’, ‘simulates’, or as we rather will say, “mimics” another derivation $\mathcal{D}$. An obvious possibility for such a stipulation is formulated in the following definition.

Definition B.1.13 (Mimicking derivations). Let $\mathcal{H}_1$ and $\mathcal{H}_2$ be APHS's, and let $\mathcal{D}_1$ and $\mathcal{D}_2$ be derivations in $\mathcal{H}_1$ and $\mathcal{H}_2$, respectively.

We say that $\mathcal{D}_1$ *mimics* $\mathcal{D}_2$ (or that $\mathcal{D}_1$ is a *mimicking derivation for* $\mathcal{D}_2$, or that $\mathcal{D}_2$ *is mimicked by* $\mathcal{D}_1$), which assertion we denote by $\mathcal{D}_1 \precsim \mathcal{D}_2$, if and only if $\mathcal{D}_1$ and $\mathcal{D}_2$ have the same conclusion and if the set of assumptions of $\mathcal{D}_1$ is a subset of the set of assumptions of $\mathcal{D}_2$; more formally,

$$\begin{aligned} \mathcal{D}_1 \precsim \mathcal{D}_2 \;\Longleftrightarrow_{\text{def}}\; & \mathsf{concl}(\mathcal{D}_1) = \mathsf{concl}(\mathcal{D}_2) \;\& \\ & \& \;\mathrm{set}(\mathsf{assm}(\mathcal{D}_1)) \subseteq \mathrm{set}(\mathsf{assm}(\mathcal{D}_2)) \end{aligned} \tag{B.7}$$

is stipulated. ☒

Two other possibilities of 'mimicking relations', the relations $\simeq^{(\mathrm{s})}$ and $\simeq^{(\mathrm{m})}$ on the class of derivations in APHS's, which demand that, apart from having the same conclusion, derivations have the same set, or respectively multiset, of assumptions, are introduced in [Gra03a]. There, also correspondences between any of these three notions of "mimics" with respective variant notions of rule derivability are established.

Apart from the notation for extensions of APHS's by adding new axioms or rules, in some statements below we will also use the following notation for APHS's that result from an APHS $\mathcal{H}$ by removing one of its named rules: for all APHS's $\mathcal{H} = \langle Fo, Na, n\mathcal{A}x, n\mathcal{R}\rangle$ and for all named rules $\boldsymbol{R} \in n\mathcal{R}$, we denote by $\mathcal{H}{-}\boldsymbol{R}$ the APHS $\langle Fo, Na, n\mathcal{A}x, n\mathcal{R}\backslash\{\boldsymbol{R}\}\rangle$.

Now we define two 'natural' abstract notions of rule elimination in APHS's.

Definition B.1.14 (Two notions of rule elimination). Let $\mathcal{H}$ be an APHS, and let $\boldsymbol{R}$ be a named rule of $\mathcal{H}$. In the two items below, we define two notions of "rule elimination".

(i) We say that $\boldsymbol{R}$*-elimination holds in* $\mathcal{H}$ if and only if

$$(\forall \mathcal{D} \in \mathcal{D}er_{\emptyset}(\mathcal{H}))\,(\exists \mathcal{D}' \in \mathcal{D}er(\mathcal{H}{-}\boldsymbol{R}))\,\big[\mathcal{D}' \precsim \mathcal{D}\big] \tag{B.8}$$

holds, i.e. iff for every derivation in $\mathcal{H}$ without assumptions there exists a mimicking derivation in $\mathcal{H}$ that does not contain applications of $\boldsymbol{R}$.

(ii) We say that $\boldsymbol{R}$*-elimination holds for* $\mathcal{D}er(\mathcal{H})$ if and only if

$$(\forall \mathcal{D} \in \mathcal{D}er(\mathcal{H}))\,(\exists \mathcal{D}' \in \mathcal{D}er(\mathcal{H}{-}\boldsymbol{R}))\,\big[\mathcal{D}' \precsim \mathcal{D}\big] \tag{B.9}$$

holds, i.e. iff for every derivation in $\mathcal{H}$ there exists a mimicking derivation in $\mathcal{H}$ without applications of $\boldsymbol{R}$.

The following theorem gives a characterization of the two notions of rule elimination defined above in terms of rule admissibility and rule derivability, respectively.

Theorem B.1.15 (Rule elimination versus rule admissibility and derivability). *Let $\mathcal{H}$ be an APHS, and let $\boldsymbol{R}$ be a named rule of $\mathcal{H}$. Then the following statements hold:*

(i) $\boldsymbol{R}$-elimination holds in $\mathcal{H}$ $\iff$ $\boldsymbol{R}$ is admissible in $\mathcal{H}{-}\boldsymbol{R}$.

(ii) $\boldsymbol{R}$-elimination holds for $\mathcal{D}er(\mathcal{H})$ $\iff$ $\boldsymbol{R}$ is derivable in $\mathcal{H}{-}\boldsymbol{R}$.

B.2 Abstract Natural-Deduction Systems

For the purpose of collecting general results about the notions of derivability and admissibility of inference rules in natural-deduction systems, we generalize and adapt the ideas that have lead to the construction of abstract pure Hilbert systems, and we thereby introduce the framework of "abstract natural-deduction system". In such systems a rule is a set of instances that is endowed with respective conclusion, premise, present-marked-assumptions and discharged-marked-assumptions functions; these functions respectively assign to an instance its conclusion, its sequence of premises, the sequence of open marked assumptions that have to be present for each premise apart, and the set of marked assumptions that get discharged by the instance. After defining this notion of ANDS-rule with respect to given sets of formulas and markers, we define abstract natural-deduction systems as structures that consist of a set of formulas, a set of assumption markers, a set of names, and a set of named rules (with names from the set of names) that 'operate' on the set of formulas of the system and use assumption markers from its set of assumption markers.

For describing our formalization of abstract natural-deduction systems in precise terms, we will need, relative to arbitrary given sets of formulas and assumption markers, the dependent notions of marked formulas, environments and assumption marker renamings. These will be defined below.

Given a set *Fo* of formulas and a set *Mk* of assumption markers, we define the set $\mathit{mFo}(\mathit{Fo},\mathit{Mk})$ of *marked formulas over Fo and Mk*, and the set $\mathit{Env}(\mathit{Fo},\mathit{Mk})$ of *environments over Fo and Mk* respectively by

$$\mathit{mFo}(\mathit{Fo},\mathit{Mk}) =_{\mathrm{def}} \{A^u \mid A \in \mathit{Fo},\, u \in \mathit{Mk}\}\ , \quad \text{and by}$$

$$\mathit{Env}(\mathit{Fo},\mathit{Mk}) =_{\mathrm{def}} \Big\{\tilde{\Sigma} \in \mathcal{P}_{\mathrm{f}}(\mathit{mFo}(\mathit{Fo},\mathit{Mk})) \mid (\forall A^{u_1}, B^{u_2} \in \tilde{\Sigma})\,[\,u_1 = u_2 \;\Rightarrow\; A = B\,]\Big\}\ .$$

Marked formulas are thus formulas with an attached assumption marker and environments are finite sets of marked formulas, in which all occurring assumption markers are different.

Let again *Fo* be a set of formulas, *Mk* a set of assumption markers, and let now *mFo* denote the set $\mathit{mFo}(\mathit{Fo},\mathit{Mk})$. We call every bijective function $r : \mathit{Mk} \to \mathit{Mk}$ an *assumption marker renaming* (or shorter, a *marker renaming*) on *Mk*. Furthermore we define, for all marker renamings r, extensions of r to marked formulas and to sets of marked formulas, that is, we define the three functions $\bar{r}_1$, $\bar{r}_2$, and $\bar{r}_3$ from r as follows:

$$\bar{r}_1 : \mathit{mFo} \to \mathit{mFo},\;\; A^u \mapsto \bar{r}_1(A^u) =_{\mathrm{def}} A^{r(u)}\ , \tag{B.10}$$

$$\bar{r}_2 : \mathcal{P}(\mathit{mFo}) \to \mathcal{P}(\mathit{mFo}),\;\; \tilde{\Sigma} \mapsto \bar{r}_2(\tilde{\Sigma}) =_{\mathrm{def}} \big\{A^{r(u)} \mid A^u \in \tilde{\Sigma}\big\}\ , \tag{B.11}$$

$$\left.\begin{aligned} &\bar{r}_3 : \mathit{Seqcs}_{\mathrm{f}}(\mathcal{P}(\mathit{mFo})) \to \mathit{Seqcs}_{\mathrm{f}}(\mathcal{P}(\mathit{mFo})) \\ &\qquad \langle\tilde{\Sigma}_1,\ldots\tilde{\Sigma}_n\rangle \mapsto \bar{r}_3(\langle\tilde{\Sigma}_1,\ldots\tilde{\Sigma}_n\rangle) =_{\mathrm{def}} \langle\bar{r}_2(\tilde{\Sigma}_1),\ldots,\bar{r}_2(\tilde{\Sigma}_n)\rangle\ . \end{aligned}\right\} \tag{B.12}$$

However, below we will drop the subscripts from the extensions $\bar{r}_1$, $\bar{r}_2$, and $\bar{r}_3$ of a marker renaming r and abbreviate both extensions of r by $\bar{r}$ because it will always be clear from the object to which such a function is applied whether the assumption markers are renamed according to r in a marked formula, in a set of marked formulas, or in a finite sequence of sets of marked formulas.

Definition B.2.1 (Rules for abstract natural-deduction systems). Let *Fo* and *Mk* be sets. An *ANDS-rule R* (*a rule R for an "abstract natural-deduction system"*)[6] on (formulas of) *Fo* and (with assumption markers of) *Mk* is a quintuple of the form $\langle \mathit{Insts}, \mathsf{prem}, \mathsf{concl}, \mathsf{pmassm}, \mathsf{dmassm} \rangle$ where

- *Insts* is a set whose elements are called the *instances of R*, and

$$\mathsf{prem} : \mathit{Insts} \to \mathit{Seqcs}_{\mathrm{f}}(\mathit{Fo}) \qquad \mathsf{concl} : \mathit{Insts} \to \mathit{Fo}$$
$$\mathsf{pmassm} : \mathit{Insts} \to \mathit{Seqcs}_{\mathrm{f}}(\mathit{Env}(\mathit{Mk}, \mathit{Fo})) \qquad \mathsf{dmassm} : \mathit{Insts} \to \mathit{Env}(\mathit{Fo}, \mathit{Mk})$$

  are functions, which, for every instance ι of R, specify the sequence of *premises* of ι, the *conclusion* of ι, the sequence of sets of *present marked assumptions* for ι (which sequence declares for each premise apart the set of assumptions on which it may depend) and the set of *discharged marked assumptions* of ι (which assumptions are discharged by the application of ι from the set of all present marked assumptions on which any of the premises of ι depends); these functions furthermore satisfy the properties explained in the following two items, namely,

- for all instances $\iota \in \mathit{Insts}$, the assertions

$$\mathrm{lg}(\mathsf{prem}(\iota)) = \mathrm{lg}(\mathsf{pmassm}(\iota)) \ , \quad \text{and} \tag{B.13}$$

$$\mathsf{dmassm}(\iota) \subseteq \bigcup_{i=1}^{\mathrm{lg}(\mathsf{pmassm}(\iota))} \mathrm{proj}_i(\mathsf{pmassm}(\iota)) \in \mathit{Env}(\mathit{Fo}, \mathit{Mk}) \tag{B.14}$$

  hold, i.e. the sequence of premises of ι and the sequence of present marked assumptions of ι have equal length, the union over all premises of ι of the present marked assumptions belonging to the respective premise is an environment over *Fo* and *Mk* that furthermore contains the present marked assumptions of ι; and

- the set *Insts* of instances of R is closed under renamings of the assumption markers in the present and discharged assumptions of an instance, that is, it holds:

$$\begin{aligned}&(\forall \iota \in \mathit{Insts})\,(\forall r \text{ marker renaming on } \mathit{Mk})\\ &\quad (\exists \iota' \in \mathit{Insts})\big[\,\mathsf{prem}(\iota') = \mathsf{prem}(\iota) \ \& \ \mathsf{concl}(\iota') = \mathsf{concl}(\iota) \ \&\\ &\qquad\qquad \& \ \mathsf{pmassm}(\iota') = \bar{r}(\mathsf{pmassm}(\iota)) \ \&\\ &\qquad\qquad \& \ \mathsf{dmassm}(\iota') = \bar{r}(\mathsf{dmassm}(\iota))\,\big] \ .\end{aligned} \tag{B.15}$$

[6] The concept of "abstract natural-deduction system" will be defined below in Definition B.2.3.

We will use the symbolic denotations Insts_R, prem_R, concl_R, pmassm_R and dmassm_R whenever we want to refer directly to the instance set, the premise, conclusion, present marked assumptions, and discharged marked assumptions functions of an ANDS-rule R, respectively. And we will use ι also as a syntactical variable for instances of ANDS-rules.

For all sets Fo and Mk, we furthermore denote by $\mathfrak{R}(\mathit{Fo}, \mathit{Mk})$ the class of ANDS-rules on Fo and Mk.

⊠

Similarly as we did so earlier in the case of APHS-rules, we define, in addition to the functions prem, concl, pmassm and dmassm associated with every ANDS-rule, a function arity and partial functions $\mathsf{prem}^{(i)}$ and $\mathsf{pmassm}^{(i)}$. For every set Fo of formulas and for every rule $R = \langle \mathit{Insts}, \mathsf{prem}, \mathsf{concl}, \mathsf{pmassm}, \mathsf{dmassm}\rangle$ on Fo, we introduce the function

$$\begin{aligned} \mathsf{arity} : \mathit{Insts} &\longrightarrow \omega \\ \iota &\longmapsto \mathsf{arity}(\iota) =_{\text{def}} \lg(\mathsf{prem}(\iota)) \ , \end{aligned}$$

and, for all $i \in \omega \backslash \{0\}$, the partial functions

$$\begin{aligned} \mathsf{prem}^{(i)} : \mathit{Insts} &\rightharpoonup \mathit{Fo} \\ \iota &\longmapsto \mathsf{prem}^{(i)}(\iota) =_{\text{def}} \mathrm{proj}_i(\mathsf{prem}(\iota)) \ , \\ \mathsf{pmassm}^{(i)} : \mathit{Insts} &\rightharpoonup \mathit{Env}(\mathit{Fo}, \mathit{Mk}) \\ \iota &\longmapsto \mathsf{pmassm}^{(i)}(\iota) =_{\text{def}} \mathrm{proj}_i(\mathsf{pmassm}(\iota)) \ . \end{aligned}$$

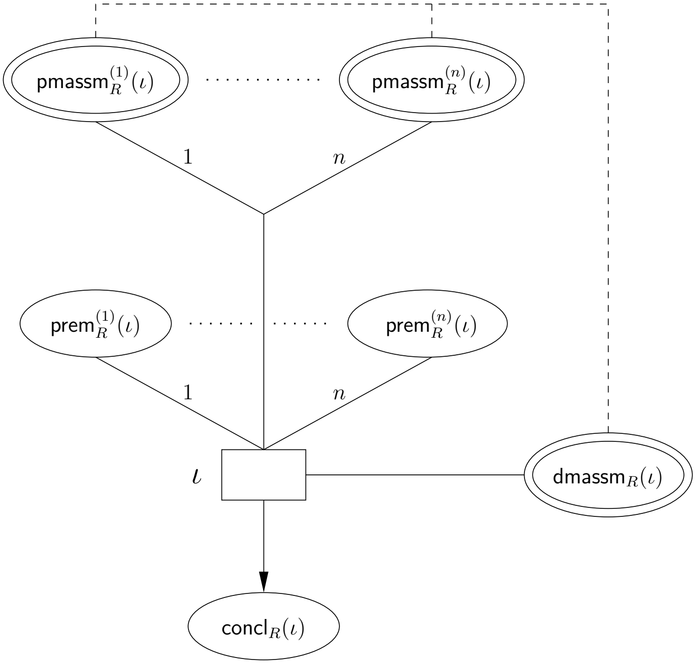
The function arity assigns to every instance ι of R the number of its premises. For all $i \in \omega\backslash\{0\}$, the partial function $\mathsf{prem}^{(i)}$ assigns to every instance ι of R its i-th premise whenever this exists; otherwise $\mathsf{prem}^{(i)}(\iota)$ is undefined. And similarly, the function $\mathsf{pmassm}^{(i)}$ assigns to every instance ι of R the environment of open marked assumptions on which the i-th premise of ι depends. We will use the denotations arity_R, $\mathsf{prem}^{(i)}_R$, and $\mathsf{pmassm}^{(i)}_R$ whenever we want to make explicit the dependence of arity, $\mathsf{prem}^{(i)}$, and $\mathsf{pmassm}^{(i)}$ upon the particular rule R to which these functions belong.

A visualization of a application (with one or more premises) of an ANDS-rule as a more generalized form of 'hyperedge' is given in Figure B.3. Below we give an example of the formalization as an ANDS-rule of a rule from a well-known natural-deduction-style proof systems for minimal, intuitionistic, and classical predicate logic.

Example B.2.2. The *exists-elimination rule* $\exists E$ of one of the systems $\mathbf{N}[\mathbf{mic}]$ from [TS00] with applications of the following form

$$\dfrac{\begin{matrix} \\ \mathcal{D}_1 \\ \exists x A \end{matrix} \qquad \begin{matrix} [A[y/x]]^u \\ \mathcal{D}_2 \\ C \end{matrix}}{C}\ \exists\mathrm{E}, u \quad \text{(if side-condition (B.16))}$$

Figure B.3: Visualization of an instance ι of an ANDS-rule R with a non-zero arity $n = \mathsf{arity}(\iota) \in \omega \backslash \{0\}$ as a generalized form of 'hyperedge' with two kinds of 'sources' (the premises and the respective sets of present marked assumptions of ι), and with a further feature (the set of assumptions discharged by (applications of) ι); the broken line is intended to allude to the fact that the discharged marked assumptions of ι are contained among the present marked assumptions of ι.


$\mathsf{pmassm}^{(1)}_R(\iota)$
$\mathsf{pmassm}^{(n)}_R(\iota)$
1
n
$\mathsf{prem}^{(1)}_R(\iota)$
$\mathsf{prem}^{(n)}_R(\iota)$
1
n
ι
$\mathsf{dmassm}_R(\iota)$
$\mathsf{concl}_R(\iota)$

where the side-condition (B.16) is defined as

$$(x \equiv y \ \vee \ y \notin \mathrm{fv}(A)) \ \& \ y \notin \mathrm{fv}(C) \ \& \\ \& \ \begin{pmatrix} u \text{ is not the marker of an open} \\ \text{marked assumption in } \mathcal{D}_1 \end{pmatrix}. \qquad \text{(B.16)}$$

This rule can be represented as an ANDS-rule on the set *Fo* of formulas of first-order predicate logic such that $R = \langle \mathit{Insts}, \mathsf{prem}, \mathsf{concl}, \mathsf{pmassm}, \mathsf{dmassm} \rangle$ with its set of instances defined according to

$$\mathit{Insts} =_{\mathrm{def}} \left\{ \langle x, y, A, C, u, \tilde{\Sigma}_1, \tilde{\Sigma}_2 \rangle \;\middle|\; \begin{array}{l} x, y \in \mathit{Var}, \quad A, C \in \mathit{Fo}, \\ u \in \mathit{Mk}, \quad \tilde{\Sigma}_1, \tilde{\Sigma}_2 \in \mathit{Env}(\mathit{Fo}, \mathit{Mk}) \\ \text{with } \tilde{\Sigma}_1 \cup \tilde{\Sigma}_2 \in \mathit{Env}(\mathit{Fo}, \mathit{Mk}), \\ \text{such that furthermore (B.18) holds} \end{array} \right\} \qquad \text{(B.17)}$$

where the condition (B.18) is defined by

$$(x \equiv y \;\vee\; y \notin \mathrm{fv}(A)) \;\;\&\;\; y \text{ is substitutible for } x \text{ in } A \;\;\&\;\; y \notin \mathrm{fv}(C) \;\;\& \\ \& \;\; \{(A[y/x])^u\} \subseteq \tilde{\Sigma}_2 \;\;\&\;\; \big[\,(A[y/x])^u \in \tilde{\Sigma}_2 \;\Rightarrow\; (\forall D^v \in \tilde{\Sigma}_1)[\,v \neq u\,]\,\big]\;, \qquad \text{(B.18)}$$

and with premise, conclusion, present marked assumption and discharged marked assumption functions of R respectively defined by

$$\mathsf{prem}(\langle x, y, A, C, u, \tilde{\Sigma}_1, \tilde{\Sigma}_2\rangle) =_{\mathrm{def}} \langle \exists x A,\, C\rangle\;,$$

$$\mathsf{concl}(\langle x, y, A, C, u, \tilde{\Sigma}_1, \tilde{\Sigma}_2\rangle) =_{\mathrm{def}} C\;,$$

$$\mathsf{pmassm}(\langle x, y, A, C, u, \tilde{\Sigma}_1, \tilde{\Sigma}_2\rangle) =_{\mathrm{def}} \langle \tilde{\Sigma}_1, \tilde{\Sigma}_2\rangle\;,$$

$$\mathsf{dmassm}(\langle x, y, A, C, u, \tilde{\Sigma}_1, \tilde{\Sigma}_2\rangle) =_{\mathrm{def}} \begin{cases} \{\,[A[y/x]]^u\} & \ldots\; (A[y/x])^u \in \tilde{\Sigma}_2 \\ \emptyset & \ldots\; \text{else}\;, \end{cases}$$

for all $\langle x, y, A, C, u, \tilde{\Sigma}_1, \tilde{\Sigma}_2\rangle \in \mathit{Insts}$. ■

Now we proceed to give the definition of “abstract natural-deduction systems”.

Definition B.2.3 (Abstract natural-deduction systems). An *abstract natural-deduction system* (an ANDS) $\mathcal{S}$ is a quadruple $\langle \mathit{Fo}, \mathit{Mk}, \mathit{Na}, n\mathcal{R}\rangle$ where

- Fo, Mk, Na and $n\mathcal{R}$ are sets whose elements are respectively called *formulas of* $\mathcal{S}$, *assumption markers of* $\mathcal{S}$ (mostly just called the *markers of* $\mathcal{S}$), *names* (for rules) *in* $\mathcal{S}$, and *named rules of* $\mathcal{S}$; we demand the set Fo to be nonempty and the set Mk to be a countably infinite set;

- $n\mathcal{R} \subseteq \mathfrak{R}(\mathit{Fo}, \mathit{Mk}) \times \mathit{Na}$, i.e. the named rules of $\mathcal{S}$ are tuples with ANDS-rules on Fo and Mk as their first, and names in $\mathcal{S}$ as their second components;

  (as done so previously for named APHS-rules, we will again use the boldface symbol $\boldsymbol{R}$, possibly with subscripts, superscripts or accents attached to it, as syntactical variable for named rules; for a named rule $\boldsymbol{R}$ we denote by $\mathsf{rule}(\boldsymbol{R})$, and by $\mathsf{name}(\boldsymbol{R})$, the first, and respectively the second component of $\boldsymbol{R}$ (the *rule* and the *name* of $\boldsymbol{R}$); furthermore we will frequently write, for example, $\boldsymbol{R} = \langle R, \mathsf{name}(\boldsymbol{R})\rangle$ and use hereby (and in other situations) the non-boldface symbol R for the rule component $\mathsf{rule}(\boldsymbol{R})$ of a named rule $\boldsymbol{R}$);

- different named rules have different names, more precisely it holds:

$$(\forall \boldsymbol{R}_1, \boldsymbol{R}_2 \in n\mathcal{R})\,\big[\,\boldsymbol{R}_1 \neq \boldsymbol{R}_2 \implies \mathsf{name}(\boldsymbol{R}_1) \neq \mathsf{name}(\boldsymbol{R}_2)\,\big]\;.$$

We denote by $\mathfrak{ND}$ the class of all abstract natural-deduction systems. If, for a some abstract natural-deduction system $\mathcal{S}$, we want to refer to its set of formulas, its set of axioms, its set of names or its set of rules, then we will use the symbolic denotations $\mathit{Fo}_{\mathcal{S}}$, $\mathit{Na}_{\mathcal{S}}$, $\mathit{Ax}_{\mathcal{S}}$ or $\mathcal{R}_{\mathcal{S}}$, respectively. And furthermore, for all ANDS's $\mathcal{S}$, we will respectively denote by $\mathit{mFo}_{\mathcal{S}}$ and by $\mathit{Env}_{\mathcal{S}}$ the set $\mathit{mFo}(\mathit{Fo}_{\mathcal{S}}, \mathit{Mk}_{\mathcal{S}})$ of *marked formulas of* $\mathcal{S}$ and the set $\mathit{Env}(\mathit{Fo}_{\mathcal{S}}, \mathit{Mk}_{\mathcal{S}})$ of *environments of* $\mathcal{S}$.

⊠

Remark B.2.4. Let $\mathcal{S}$ be an ANDS, and let $\boldsymbol{R} = \langle R, \mathsf{name}(\boldsymbol{R})\rangle$ be a named rule of $\mathcal{S}$.

The operational behaviour of an instance $\iota \in \mathit{Insts}_R$ with $\mathsf{arity}(\iota) = n \in \omega\backslash\{0\}$ can be expressed in a sequent-style formalism as an inference of the following kind:

$$\frac{\mathsf{pmassm}_R^{(1)}(\iota) \;\Rightarrow\; \mathsf{prem}_R^{(1)}(\iota) \qquad \ldots \qquad \mathsf{pmassm}_R^{(n)}(\iota) \;\Rightarrow\; \mathsf{prem}_R^{(n)}(\iota)}{\left(\bigcup_{i=1}^{n} \mathsf{pmassm}_R^{(i)}(\iota)\right) \setminus \mathsf{dmassm}_R(\iota) \;\Rightarrow\; \mathsf{concl}_R(\iota)}\ \mathsf{name}(\boldsymbol{R})$$

And in a natural-deduction-style formalization of derivations, an instance $i \in \mathit{Insts}_R$ with $\mathsf{arity}(\iota) = n \in \omega\backslash\{0\}$ gives rise to inferences like the one at the bottom of the derivation $\mathcal{D}$ that is indicated as the symbolic prooftree

$$\frac{\begin{array}{c}\{[\mathsf{pmassm}_R^{(1)}(\iota)]\}\\ \mathcal{D}_1\\ \mathsf{prem}_R^{(1)}(\iota)\end{array} \qquad \ldots \qquad \begin{array}{c}\{[\mathsf{pmassm}_R^{(n)}(\iota)]\}\\ \mathcal{D}_n\\ \mathsf{prem}_R^{(n)}(\iota)\end{array}}{\mathsf{concl}_R(\iota)}\ \mathsf{name}(\boldsymbol{R}),\ u_1, \ldots, u_m$$

where $u_1, \ldots, u_m$ are such that $\mathsf{dmassm}(\iota) = \{D_1^{u_1}, \ldots, D_m^{u_m}\}$ for some $m \in \omega$ and marked formulas $D_1^{u_1}, \ldots, D_m^{u_m} \in \mathit{mFo}_{\mathcal{S}}$, and where at the top the designations of the form $\{[\mathsf{pmassm}_R^{(1)}(\iota)]\}$, ..., $\{[\mathsf{pmassm}_R^{(n)}(\iota)]\}$ stand for the sets $\mathsf{pmassm}_R^{(1)}(\iota)$, ..., $\mathsf{pmassm}_R^{(n)}(\iota)$ of open assumption classes in $\mathcal{D}_1, \ldots, \mathcal{D}_n$, respectively.

We continue by introducing the notion of "derivation" in an ANDS, in a straightforward, albeit formally involved, way.

Definition B.2.5 (Derivations in ANDS's). Let $\mathcal{S} = \langle \mathit{Fo}, \mathit{Mk}, \mathit{Na}, n\mathcal{R}\rangle$ be an ANDS.

The set $\mathcal{D}er(\mathcal{S})$ of the *derivations in* $\mathcal{S}$ is the set of prooftrees that are generated by carrying out a finite number of construction steps of the kinds (i) and (ii) described below[7]. Simultaneously with this inductive definition also the three functions

$$\mathsf{omassm}\colon \mathcal{D}er(\mathcal{S}) \to \mathcal{P}(\mathit{mFo}) \qquad \mathsf{concl}\colon \mathcal{D}er(\mathcal{S}) \to \mathit{Fo} \qquad |\cdot|\colon \mathcal{D}er(\mathcal{S}) \to \omega$$

are defined that respectively assign to a derivation $\mathcal{D}$ in $\mathcal{S}$ the set $\mathsf{omassm}(\mathcal{D})$ of *open (marked) assumptions* of $\mathcal{D}$, the *conclusion* $\mathsf{concl}(\mathcal{D})$ of $\mathcal{D}$, and the *(rule application) depth* $|\mathcal{D}|$ of $\mathcal{D}$.

(i) For all formulas $A \in \mathit{Fo}$ and markers $u \in \mathit{Mk}$, the prooftree $\mathcal{D}$ consisting only of the marked formula

$$A^u \tag{B.19}$$

is a derivation in $\mathcal{S}$ with open marked assumptions $\mathsf{omassm}(\mathcal{D}) =_{\mathrm{def}} \{A^u\}$, with conclusion $\mathsf{concl}(\mathcal{D}) =_{\mathrm{def}} A$, and with depth $|\mathcal{D}| =_{\mathrm{def}} 0$.

[7] Equivalently, $\mathcal{D}er(\mathcal{S})$ is the smallest set that is closed under these generation steps (i) and (ii).

(ii) Let a named rule $\boldsymbol{R} = \langle R, \mathsf{name}(\boldsymbol{R})\rangle \in n\mathcal{R}$ and an instance $\iota \in \mathit{Insts}_R$ be given. We distinguish two cases concerning the arity of ι:

Case 1. $\mathsf{arity}_R(\iota) = 0$.
Given that $\mathsf{concl}_R(\iota) = A$, the prooftree

$$\frac{}{A}\,\mathsf{name}(\boldsymbol{R}) \tag{B.20}$$

is a derivation $\mathcal{D}$ in $\mathcal{S}$ with conclusion $\mathsf{concl}(\mathcal{D}) =_{\text{def}} A$ and without open assumptions, i.e. with $\mathsf{omassm}(\mathcal{D}) =_{\text{def}} \emptyset$; its depth is $|\mathcal{D}| =_{\text{def}} 1$. In this situation we say that $\mathcal{D}$ *results by an application of* ι *from the empty sequence of derivations in* $\mathcal{S}$.

Case 2. $\mathsf{arity}_R(\iota) = n \in \omega\backslash\{0\}$.
For all derivations $\mathcal{D}_1, \ldots, \mathcal{D}_n \in \mathcal{D}er(\mathcal{S})$, $m \in \omega$, markers $u_1, \ldots, u_m \in$ $\in \mathit{Mk}$, and formulas $A, A_1, \ldots, A_n$, $D_1, \ldots, D_m \in \mathit{Fo}$ such that the conditions

$$\mathsf{prem}_R(\iota) = \langle A_1, \ldots, A_n\rangle\ , \qquad \mathsf{concl}_R(\iota) = A\ ,$$

$$\mathsf{pmassm}_R^{(i)}(\iota) \;=\; \mathsf{omassm}(\mathcal{D}_i) \quad \text{(for all } 1 \le i \le n)\ ,$$

$$\mathsf{dmassm}_R(\iota) = \{D_1^{u_1}, \ldots, D_m^{u_m}\}\ , \ \text{ and}$$

$$\mathsf{concl}(\mathcal{D}_i) = A_i \quad \text{(for all } 1 \le i \le n)$$

are fulfilled, the prooftree of the form

$$\frac{\begin{matrix}\mathcal{D}_1 & & \mathcal{D}_n\\ A_1 & \ldots & A_n\end{matrix}}{A}\,\mathsf{name}(\boldsymbol{R}),\, u_1, \ldots, u_m \tag{B.21}$$

is a derivation $\mathcal{D}$ in $\mathcal{S}$ with marked open assumptions, conclusion and depth that are respectively defined by

$$\mathsf{concl}(\mathcal{D}) \;=_{\text{def}}\; \mathsf{concl}_R(\iota) = A\ ,$$

$$\mathsf{omassm}(\mathcal{D}) \;=_{\text{def}}\; \Big(\bigcup_{i=1}^{n} \mathsf{omassm}(\mathcal{D}_i)\Big) \setminus \mathsf{dmassm}_R(\iota)\ ,$$

$$|\mathcal{D}| \;=_{\text{def}}\; 1 + \max\{\,|\mathcal{D}_i| \mid i \in \omega,\ 1 \le i \le n\}\ .$$

In this case we say that $\mathcal{D}$ *results from the sequence* $\langle\mathcal{D}_1, \ldots, \mathcal{D}_n\rangle$ *of derivations in* $\mathcal{S}$ *by an application of* ι.

Furthermore we define by

$$\mathcal{D}er_{\emptyset}(\mathcal{S}) \;=_{\text{def}}\; \{\mathcal{D} \in \mathcal{D}er(\mathcal{S}) \mid \mathsf{omassm}(\mathcal{D}) = \emptyset\}\ .$$

the set of all derivations in $\mathcal{S}$ without open assumptions. ⊠

The following lemma formulates the easy observation about the definition above that the open marked assumptions of derivations in an ANDS are in fact environments.

Proposition B.2.6. *Let $\mathcal{S}$ be an ANDS. Then it holds:*

$$(\forall \mathcal{D} \in \mathcal{D}er(\mathcal{S}))\left[\,\mathsf{omassm}(\mathcal{D}) \in \mathit{Env}_{\mathcal{S}}\,\right] .$$

Sketch of Proof. This can be shown by straightforward induction on the depth $|\mathcal{D}|$ of derivations $\mathcal{D}$ in $\mathcal{S}$. In the induction step the condition (B.14) on the union of the present marked assumptions of a rule application in $\mathcal{S}$ (to be an environment of $\mathcal{S}$) is used as well as the simple fact that subsets of environments are environments. □

The next lemma formulates an easy consequence that condition (B.15) in the definition of ANDS-rules has for the set of derivations in an ANDS. While (B.15) demands of the set of instances of an ANDS-rule to be closed under renamings of markers in the present and discharged marked assumptions of an instance, the lemma below states that the set of derivations in $\mathcal{S}$ is closed under renamings of assumption markers in derivations.

For its formulation, we define applications of assumption marker renamings to derivations: Let $\mathcal{S}$ be an ANDS, and let r be a marker renaming on $\mathit{Mk}_{\mathcal{S}}$. Then we define, next to the extensions $\bar{r}_1$ and $\bar{r}_2$ of r as introduced in (B.10) and (B.11), the extension

$$\bar{r}_4 : \mathcal{D}er(\mathcal{S}) \rightarrow \mathcal{D}er(\mathcal{S}),\ \ \mathcal{D} \mapsto \bar{r}_4(\mathcal{D}) ,$$

of r to derivations of $\mathcal{S}$ by specifying, for all $\mathcal{D} \in \mathcal{D}er(\mathcal{S})$, *the renaming $\bar{r}_4(\mathcal{D})$ of $\mathcal{D}$ with respect to r* as the prooftree that is the result of simultaneously renaming all occurrences of an assumption marker u in $\mathcal{D}$ (where $u \in \mathit{Mk}$) by $r(u)$, respectively. Similarly as in the case of the two other extensions of marker renamings, we will drop the subscript from extensions $\bar{r}_4$ of marker renamings r and just write $\bar{r}$ instead since it will always be clear from the surrounding (for example, textual) context which sort of extensions of marker renamings is meant.

The following proposition states that the role of assumption markers in derivations of an ANDS is indeed restricted to that of an auxiliary feature (of binding open assumptions to rule applications and of providing names for different assumption classes). It states that the set of derivations in an ANDS is closed under marker renamings, and that the application of a marker renaming r to a derivation $\mathcal{D}$ in an ANDS $\mathcal{S}$ results in a derivation $\bar{r}(\mathcal{D})$ in $\mathcal{S}$ with the same conclusion as $\mathcal{D}$ and with open assumption classes that respectively arise by renaming the assumption classes of $\mathcal{D}$ according to r.

Proposition B.2.7 (Renaming of assm. markers in ANDS-derivations). *Let $\mathcal{S}$ be an ANDS. And furthermore, let $\mathcal{D}$ be a derivation in $\mathcal{S}$, and r a marker renaming on $\mathit{Mk}_{\mathcal{S}}$. The following two assertions hold for the renaming $\bar{r}(\mathcal{D})$ of $\mathcal{D}$ with respect to r:*

(i) $\bar{r}(\mathcal{D}) \in \mathcal{D}er(\mathcal{S})$.

(ii) $\mathsf{omassm}(\bar{r}(\mathcal{D})) = \bar{r}(\mathsf{omassm}(\mathcal{D}))$ *and* $\mathsf{concl}(\bar{r}(\mathcal{D})) = \mathsf{concl}(\mathcal{D})$.

Sketch of Proof. For every ANDS $\mathcal{S}$ and every marker renaming r on $Mk_{\mathcal{S}}$, the fulfilledness, for all $\mathcal{D} \in \mathcal{D}er(\mathcal{S})$, of assertions (i) and (ii) of the lemma can be shown by a straightforward proof using induction on $|\mathcal{D}|$; this proof relies on the property (B.15) of ANDS-rules in an essential way. □

Lemma B.2.7 obviously implies the following: For every derivation $\mathcal{D}$ in an ANDS $\mathcal{S}$ and for every marker renaming r on $Mk_{\mathcal{S}}$ there exists a derivation $\mathcal{D}'$ in $\mathcal{S}$ with the same conclusion as $\mathcal{D}$ and such that the set of open marked assumptions of $\mathcal{D}'$ results from the set of open marked assumptions of $\mathcal{D}$ by renaming assumption markers according to r. More formally, for all ANDS's $\mathcal{S}$ it holds:

$$\begin{aligned}&(\forall \mathcal{D} \in \mathcal{D}er(\mathcal{S}))\,(\forall r \text{ marker renaming on } Mk_{\mathcal{S}})\\ &\qquad (\exists \mathcal{D}' \in \mathcal{D}er(\mathcal{S}))\,\big[\,\mathsf{omassm}(\mathcal{D}') = \bar{r}(\mathsf{omassm}(\mathcal{D}))\;\;\&\\ &\qquad\qquad\qquad\qquad\qquad\qquad\&\;\;\mathsf{concl}(\mathcal{D}') = \mathsf{concl}(\mathcal{D})\,\big]\,. \qquad \text{(B.22)}\end{aligned}$$

In the next definition we introduce a general notion of consequence relation in an ANDS which relates sets of marked assumptions with single formulas. For an ANDS $\mathcal{S}$, the consequence relation $\vdash_{\mathcal{S}}$ is defined such that, for all sets $\tilde{\Sigma}$ of marked formulas and all formulas A, $\tilde{\Sigma} \vdash_{\mathcal{S}} A$ holds if and only if there is a derivation $\mathcal{D}$ in $\mathcal{S}$ with open marked assumptions $\tilde{\Sigma}$ and with conclusion A.

We note that, compared with the 'standard' consequence relation in APHS's (see Definition B.1.4), the consequence relation defined here in ANDS's is stricter in the sense that, for a considered ANDS $\mathcal{S}$, $\tilde{\Sigma} \vdash_{\mathcal{S}} A$ holds only if there exists a deduction of A in $\mathcal{S}$ that uses *all* of the (marked) assumptions in $\tilde{\Sigma}$ *at least once*. This form of consequence relation is more appropriate for ANDS's owing to the following fact: contrary to the situation in APHS's, for a rule R of an ANDS $\mathcal{S}$ to be applicable to a sequence $\langle \mathcal{D}_1, \ldots, \mathcal{D}_n \rangle$ of subderivations in $\mathcal{S}$ with resulting derivation $\mathcal{D}$ it is typically necessary to have full knowledge of the open marked assumptions $\mathsf{omassm}(\mathcal{D}_1), \ldots, \mathsf{omassm}(\mathcal{D}_n)$ in the immediate subderivations $\mathcal{D}_1, \ldots, \mathcal{D}_n$ of $\mathcal{D}$.

Definition B.2.8 (A consequence relation on ANDS's). Let $\mathcal{S}$ be an ANDS with set Fo of formulas and with set mFo of marked formulas. We define the consequence relation $\vdash_{\mathcal{S}}$ on $\mathcal{S}$, where $\vdash_{\mathcal{S}} \subseteq \mathcal{P}(mFo) \times Fo$, by stipulating for all $A \in Fo$ and *sets* $\tilde{\Sigma} \in \mathcal{P}(mFo)$

$$\tilde{\Sigma} \vdash_{\mathcal{S}} A \;\;\Longleftrightarrow_{\text{def}}\;\; (\exists \mathcal{D} \in \mathcal{D}er(\mathcal{S}))\,\big[\,\mathsf{omassm}(\mathcal{D}) = \tilde{\Sigma} \;\;\&\;\; \mathsf{concl}(\mathcal{D}) = A\,\big]\;; \qquad \text{(B.23)}$$

if, for some $A \in Fo$ and $\tilde{\Sigma} \in \mathcal{P}(mFo)$, $\tilde{\Sigma} \vdash_{\mathcal{S}} A$ holds, then we say that A *can be derived in* $\mathcal{S}$ *from the marked assumptions in* $\tilde{\Sigma}$. ☒

The next proposition formulates the obvious consequence of (B.22) that the consequence relation $\vdash_{\mathcal{S}}$ is not 'sensitive' to the particular assumption markers used in marked assumptions. This statement can be read as a further confirmation that the role of assumption markers as part of the concept of ANDS is indeed only that of auxiliary devices for bookkeeping purposes (concerning the assumption-discharging feature of ANDS-rules).

Proposition B.2.9. *Let $\mathcal{S}$ be an ANDS with set Fo of formulas and with set mFo of marked formulas. The consequence relation $\vdash_{\mathcal{S}}$ is invariant under marker renamings applied to marked assumptions, that is,*

$$\tilde{\Sigma} \vdash_{\mathcal{S}} A \iff \bar{r}(\tilde{\Sigma}) \vdash_{\mathcal{S}} A$$

holds for all $\tilde{\Sigma} \in \mathcal{P}(\mathcal{S})$ and $A \in Fo$.

We carry on by giving definitions for the notions of "theorem" in an ANDS, of the "theory" of an ANDS, and of two ANDS's being equivalent. And in the subsequent definition we adapt the notion "extension by enlargement" to ANDS's.

Definition B.2.10. (Theorems, theory of an ANDS; equivalent ANDS's). Let $\mathcal{S}$ be an ANDS. A formula $A \in Fo_{\mathcal{S}}$ is a *theorem* of $\mathcal{S}$ if and only if $\emptyset \vdash_{\mathcal{S}} A$, i.e. iff there exists a derivation $\mathcal{D}$ in $\mathcal{S}$ from the empty set of open assumptions and with conclusion A; in this case we write $\vdash_{\mathcal{S}} A$ for $\emptyset \vdash_{\mathcal{S}} A$. The *theory of* $\mathcal{S}$ is the set $Th(\mathcal{S}) = \{ A \in Fo_{\mathcal{S}} \mid \vdash_{\mathcal{S}} A \}$ of theorems of $\mathcal{S}$.

Let $\mathcal{S}_1$ and $\mathcal{S}_2$ be ANDS's. We say that $\mathcal{S}_1$ and $\mathcal{S}_2$ are *equivalent* (which is denoted by $\mathcal{S}_1 \sim_{th} \mathcal{S}_2$) if and only if $Th(\mathcal{S}_1) = Th(\mathcal{S}_2)$ holds. ⊠

Definition B.2.11 (Extensions by enlargement of ANDS's). For the following stipulations, let $\mathcal{S} = \langle Fo, Mk, Na, n\mathcal{R} \rangle$ be an ANDS.

(i) An *extension by enlargement of* $\mathcal{S}$ is an ANDS $\mathcal{S}' = \langle Fo', Mk', Na', n\mathcal{R}' \rangle$ that results from $\mathcal{S}$ by extending the sets of formulas, of names, of named axioms and/or of named rules of $\mathcal{S}$, i.e. iff the following holds:

$$Fo \subseteq Fo' \;\;\&\;\; Mk \subseteq Mk' \;\;\&\;\; Na \subseteq Na' \;\;\&\;\; n\mathcal{R} \subseteq n\mathcal{R}' \,.$$

(ii) Let R be an ANDS-rule on Fo on Mk. An *extension of* $\mathcal{S}$ *by adding* R *as a new rule* (or shorter, an *extension of* $\mathcal{S}$ *by adding the new rule* R) is an extension by enlargement of the form

$$\mathcal{S}' = \langle Fo, Mk, Na', n\mathcal{R} \cup \{\langle R, name \rangle\} \rangle$$

for some $name \in Na'$ such that $\langle R, name \rangle \notin n\mathcal{R}$ holds. An extension of $\mathcal{S}$ by adding R as a new rule will generally be denoted by $\mathcal{S}^{(+R)}$.

⊠

In analogy with Proposition B.1.7 concerning APHS's, the following proposition contains the assertion that, for all ANDS's $\mathcal{S}$ and rules R on $Fo_{\mathcal{S}}$ and $Mk_{\mathcal{S}}$, the specific chosen name of an added named version of R in an extension $\mathcal{S}^{(+R)}$ of $\mathcal{S}$ by adding the new rule R does not matter for the consequence relation $\vdash_{\mathcal{S}^{(+R)}}$ on $\mathcal{S}^{(+R)}$.

Proposition B.2.12. *Let $\mathcal{S}$ be an ANDS, and let R be an ANDS-rule on $Fo_{\mathcal{S}}$. Then the two statements*

$$\big(\forall\, \mathcal{S}_1, \mathcal{S}_2, \textit{ extensions of } \mathcal{S} \textit{ by adding the new rule } R\big) \big[\, \mathcal{S}_1 \sim_{th} \mathcal{S}_2 \,\big] \,, \tag{B.24}$$

$$\big(\forall\, \mathcal{S}_1, \mathcal{S}_2, \textit{ extensions of } \mathcal{S} \textit{ by adding the new rule } R\big) \big[\vdash_{\mathcal{S}_1} \;=\; \vdash_{\mathcal{S}_2} \big] \tag{B.25}$$

hold, i.e. all extensions $\mathcal{S}^{(+R)}$ of $\mathcal{S}$ by adding R as a new rule are equivalent, and the consequence relations that are respectively defined in $\mathcal{S}_1$ and in $\mathcal{S}_2$ according to Definition B.2.8 are equal.

Sketch of Proof. Obviously, (B.24) follows directly from (B.25). (B.24) can be shown, by letting $\mathcal{S}_1$ and $\mathcal{S}_2$ be arbitrary extensions of $\mathcal{S}$ by adding R as a new rule, and by showing that for every derivation $\mathcal{D}_1$ in $\mathcal{S}_1$ there exists a derivation $\mathcal{D}_2$ in $\mathcal{S}_2$ with the same conclusion and with the same open assumptions, and vice versa. Now we let $\boldsymbol{R}_1$ and $\boldsymbol{R}_2$ be those named versions of R that are respectively added to $\mathcal{S}$ in $\mathcal{S}_1$ and in $\mathcal{S}_2$. To show the first half of the mentioned statement, structural induction on derivations in $\mathcal{S}_1$ can be applied to prove that, for every derivation $\mathcal{D}_1$ in $\mathcal{S}_1$, the result $\mathcal{D}_1^{[\boldsymbol{R}_2/\boldsymbol{R}_1]}$ of replacing in $\mathcal{D}_1$ each inference induced by an application of $\boldsymbol{R}_1$ by an inference induced by $\boldsymbol{R}_2$ (by merely changing the rule name labels of all inferences induced by $\boldsymbol{R}_1$ from $\mathsf{name}(\boldsymbol{R}_1)$ to $\mathsf{name}(\boldsymbol{R}_2)$) is a derivation in $\mathcal{S}_2$ with the same conclusion and with the same set of open assumptions as $\mathcal{D}_1$. □

For the definition later of rule derivability in an ANDS, we will need the auxiliary notions “pseudo-derivation” and “pseudo-derivation context” in an ANDS, which are defined next. Intuitively, a “pseudo-derivation” in an ANDS $\mathcal{S}$ is a prooftree $\mathcal{PD}$ formed with formulas of $\mathcal{S}$ and with names of rules of $\mathcal{S}$ as application labels, where the ‘inference steps’ in $\mathcal{PD}$ do not necessarily correspond (in the way described in clause (ii) of Definition B.2.5) to applications of rules of $\mathcal{S}$; yet, the condition is retained that the open assumptions of pseudo-derivations are environments (like this is the case for derivations, cf. Proposition B.2.6). And a “pseudo-derivation context” in an ANDS $\mathcal{S}$ can be viewed as the result of respectively replacing some sub-prooftrees in a pseudo-derivation in $\mathcal{S}$ by context-holes $[\,]_i$ that carry numbers $i \in \omega\backslash\{0\}$ (a hole $[\,]_i$, for $i \in \omega\backslash\{0\}$, may respectively occur in a “pseudo-derivation context” an arbitrary (finite) number of times).

In the definition below, however, pseudo-derivation contexts are defined first, and then pseudo-derivations are introduced as pseudo-derivation contexts that do not contain context holes and that fulfill an additional requirement about their open assumptions.

Definition B.2.13 (Pseudo-derivation contexts, and pseudo-derivations in ANDS’s). Let $\mathcal{S} = \langle \mathit{Fo}, \mathit{Mk}, \mathit{Na}, \mathit{n}\mathcal{R}\rangle$ be an ANDS, and let $k \in \omega$.
The set $\mathcal{PD}\mathit{erCtxt}_k(\mathcal{S})$ of *k-ary pseudo-derivation contexts in $\mathcal{S}$* is defined as the smallest set (eventually consisting of prooftrees) that is closed under the four generation steps (i), (ii), (iii), and (iv) that are given below. Simultaneously with this definition also the three functions

$$\mathsf{omassm} : \mathcal{PD}\mathit{erCtxt}_k(\mathcal{S}) \to \mathcal{M}_{\mathrm{f}}(\mathit{Fo}) \qquad |\cdot| : \mathcal{PD}\mathit{erCtxt}_k(\mathcal{S}) \to \omega$$
$$\mathsf{concl} : \mathcal{PD}\mathit{erCtxt}_k(\mathcal{S}) \to \mathit{Fo} \cup \{\, [\,]_i \mid i \in \omega,\, 1 \le i \le k\}$$

are defined that respectively assign to a k-ary pseudo-derivation context $\mathcal{PC}$ in $\mathcal{S}$ the set $\mathsf{omassm}(\mathcal{PC})$ of *open marked assumptions* of $\mathcal{PC}$, the *conclusion* $\mathsf{concl}(\mathcal{PC})$ of $\mathcal{PC}$, and the *(rule application) depth* $|\mathcal{PC}|$. As done so here already, we will use the concatenated letters $\mathcal{PC}$ for syntactical variables of pseudo-derivation contexts.

(i) For every $i \in \{1, \ldots, k\}$, the prooftree (consisting only of the i-th hole $[\,]_i$)

$$[\,]_i \tag{B.26}$$

is a k-ary pseudo-derivation context $\mathcal{PC}$ in $\mathcal{S}$ without open assumptions, i.e. with $\mathsf{omassm}(\mathcal{PC}) =_{\text{def}} \emptyset$, with conclusion $\mathsf{concl}(\mathcal{PC}) =_{\text{def}} [\,]_i$, and with depth $|\mathcal{PC}| =_{\text{def}} 0$.

(ii) For all formulas $A \in \mathit{Fo}$ and markers $u \in \mathit{Mk}$, the prooftree $\mathcal{PC}$ of the form

$$A^u \tag{B.27}$$

is a k-ary pseudo-derivation context in $\mathcal{S}$ with conclusion $\mathsf{concl}(\mathcal{PC}) =_{\text{def}} A$, with open assumptions $\mathsf{omassm}(\mathcal{PC}) =_{\text{def}} \{A^u\}$, and with depth $|\mathcal{PC}| =_{\text{def}} 0$.

(iii) For all named rules $\boldsymbol{R} \in n\mathcal{R}$, and for all formulas $A \in \mathit{Fo}$, the prooftree

$$\frac{}{A}\ \mathsf{name}(\boldsymbol{R}) \tag{B.28}$$

is a k-ary derivation context in $\mathcal{S}$ with $\mathsf{omassm}(\mathcal{PC}) =_{\text{def}} \emptyset$ (i.e. without open assumptions), with conclusion $\mathsf{concl}(\mathcal{PC}) =_{\text{def}} A$, and with $|\mathcal{PC}| =_{\text{def}} 1$.

(iv) For all named rules $\boldsymbol{R} \in n\mathcal{R}$, natural numbers $n \in \omega\backslash\{0\}$, k-ary pseudo-derivation contexts $\mathcal{PC}_1, \ldots, \mathcal{PC}_n$ in $\mathcal{S}$, formulas $A \in \mathit{Fo}$, and for all $m \in \omega$ assumption markers $u_1, \ldots, u_m \in \mathit{Mk}$, the prooftree of the form

$$\frac{\mathcal{PC}_1 \qquad \ldots \qquad \mathcal{PC}_n}{A}\ \mathsf{name}(\boldsymbol{R}),\ u_1, \ldots, u_m \tag{B.29}$$

is a pseudo-derivation context $\mathcal{PC}$ in $\mathcal{S}$ with conclusion $\mathsf{concl}(\mathcal{PC}) =_{\text{def}} A$, and with open assumptions and depth defined according to

$$\begin{aligned}\mathsf{omassm}(\mathcal{PC}) \ &=_{\text{def}} \ \Big(\bigcup_{i=1}^{n} \mathsf{omassm}(\mathcal{PC}_i)\Big) \setminus \{\, A^{u_i} \mid 1 \le i \le m,\ A \in \mathit{Fo} \}\ , \\ |\mathcal{PC}| \ &=_{\text{def}} \ 1 + \max\{\, |\mathcal{PC}_i| \mid i \in \omega,\ 1 \le i \le n\}\ .\end{aligned}$$

By $\mathcal{PD}\mathit{er}\mathcal{C}\mathit{txt}(\mathcal{S})$ we designate the set of all pseudo-derivation contexts, i.e. the union of the sets $\mathcal{PD}\mathit{er}\mathcal{C}\mathit{txt}_k(\mathcal{S})$, over all $k \in \omega$.

By a *pseudo-derivation in* $\mathcal{S}$ we understand a 0-ary pseudo-derivation context $\mathcal{PC}$ in $\mathcal{S}$ (i.e. $\mathcal{PC} \in \mathcal{PD}\mathit{er}\mathcal{C}\mathit{txt}_0(\mathcal{S})$) with $\mathsf{omassm}(\mathcal{PC}) \in \mathit{Env}(\mathit{Fo}_{\mathcal{S}}, \mathit{Mk}_{\mathcal{S}})$, i.e. with the additional property that its set of open marked assumptions is an environment. We designate by $\mathcal{PD}\mathit{er}(\mathcal{S})$ the set of pseudo-derivations in $\mathcal{S}$, and we will use the connected letters $\mathcal{PD}$ for syntactical variables referring to pseudo-derivations.

⊠

Remark B.2.14. Our definition of "pseudo-derivation context" was chosen in such a way that this notion is closed under an operation of "hole-filling" (see Definition B.2.17 below).

The condition we have imposed on pseudo-derivations that their sets of open marked assumptions be environments is not strictly necessary for the following definitions and results. However, it is surely a sensible restriction to demand that for every pseudo-derivation $\mathcal{PD}$ in an ANDS $\mathcal{S}$ there exists a *derivation* $\mathcal{D}$ in some, possibly different, ANDS $\mathcal{S}'$ such that $\mathcal{D}$ and $\mathcal{PD}$ have the same conclusion and the same open assumptions; this would clearly not be the case in general if the set of open assumptions of pseudo-derivations did not have to be environments (cf. Proposition B.2.6).

For other purposes it might perhaps also be reasonable to demand that also all sub-prooftrees of pseudo-derivations have environments as their sets of open assumptions; although it is easy to give a definition for such a stronger notion of "pseudo-derivation" along the lines of the definition of pseudo-derivation contexts in Definition B.2.13, we have not introduced such a stronger notion because we do not need it here.

An obviously desirable property of the notion "pseudo-derivation" in an ANDS is formulated by following proposition, which is an easy consequence of the definitions of derivations and of pseudo-derivations in an ANDS.

Proposition B.2.15. *Let $\mathcal{S}$ be an ANDS. Every derivation in $\mathcal{S}$ is also a pseudo-derivation in $\mathcal{S}$, and consequently $\mathcal{D}er(\mathcal{S}) \subseteq \mathcal{PD}er(\mathcal{S})$ is the case.*

We carry on by defining the later important notion of a pseudo-derivation resulting from a sequence of pseudo-derivations by the application of a named rule from an ANDS.

Definition B.2.16 (Pseudo-derivations resulting by rule applications). Let $\mathcal{S} = \langle Fo, Mk, Na, n\mathcal{R}\rangle$ be an ANDS. And furthermore, let $\boldsymbol{R}$ be a named rule of $\mathcal{S}$, $n \in \omega$, and let $\mathcal{PD}, \mathcal{PD}_1, \ldots, \mathcal{PD}_n$ be pseudo-derivations in $\mathcal{S}$.

Let $\iota \in Insts_R$, where $R = \mathsf{rule}(\boldsymbol{R})$. We say that the pseudo-derivation $\mathcal{PD}$ *results from* (the sequence) $\langle \mathcal{PD}_1, \ldots, \mathcal{PD}_n\rangle$ (of pseudo-derivations) *by the application* ι of R (or just that $\mathcal{PD}$ *results from* $\langle \mathcal{PD}_1, \ldots, \mathcal{PD}_n\rangle$ *by applying* ι) if and only if $\mathcal{PD}$ is of the form

$$\frac{\mathcal{PD}_1 \qquad \ldots \qquad \mathcal{PD}_n}{A}\,\mathsf{name}(\boldsymbol{R})$$

(in the case $n = 0$ this prooftree is of the form (B.28)) and if the inference at the bottom of $\mathcal{PD}$ is performed according to ι, more precisely, if the conditions

$$\langle \mathsf{concl}(\mathcal{PD}_1), \ldots, \mathsf{concl}(\mathcal{PD}_n)\rangle = \mathsf{prem}_R(\iota)\ ,$$

$$\langle \mathsf{omassm}(\mathcal{PD}_1), \ldots, \mathsf{omassm}(\mathcal{PD}_n)\rangle = \mathsf{pmassm}_R(\iota)\ ,$$

$$\mathsf{omassm}(\mathcal{PD}) = \Big(\bigcup_{i=1}^{n} \mathsf{pmassm}_R(\iota)\Big) \setminus \mathsf{dmassm}_R(\iota)\ ,$$

$$\mathsf{concl}(\mathcal{PD}) = A = \mathsf{concl}_R(\iota)$$

are met.

And we consequently we also say that the pseudo-derivation $\mathcal{PD}$ *results from* (the sequence) $\langle \mathcal{PD}_1, \ldots, \mathcal{PD}_n \rangle$ (of pseudo-derivations) *by an application of* R if and only if there exists an instance $\iota \in \mathit{Insts}_R$ with $n = \mathsf{arity}(\iota)$, where $R = \mathsf{rule}(\boldsymbol{R})$, such that $\mathcal{PD}$ results from $\langle \mathcal{PD}_1, \ldots, \mathcal{PD}_n \rangle$ by applying ι. ⊠

Next, we define hole-filling in pseudo-derivation contexts of an ANDS: we give an inductive definition of the result $\mathcal{PC}[\tilde{\mathcal{PC}}_1, \ldots, \tilde{\mathcal{PC}}_k]$ of substituting pseudo-derivation contexts $\tilde{\mathcal{PC}}_1, \ldots, \tilde{\mathcal{PC}}_k$ for the respective occurrences of the holes $[\,]_1, \ldots, [\,]_k$ in a k-ary pseudo-derivation context $\mathcal{PC}$.

Definition B.2.17 (Hole-Filling in pseudo-derivation contexts). Let $\mathcal{S}$ be an ANDS of the form $\langle \mathit{Fo}, \mathit{Mk}, \mathit{Na}, n\mathcal{R} \rangle$.

For all $k \in \omega$, and for all $l_1, \ldots, l_k \in \omega$ and $l = \max_{1 \le i \le k} l_i$, we define, for every pseudo-derivation context $\mathcal{PC} \in \mathcal{PD}\mathit{er}\mathcal{C}\mathit{txt}_k(\mathcal{S})$, the *hole-filling operation* on $\mathcal{PC}$ as the function

$$\begin{aligned} \mathcal{PC}[\underbrace{\cdot, \ldots, \cdot}_{k}] : \mathcal{PD}\mathit{er}\mathcal{C}\mathit{txt}_{l_1}(\mathcal{S}) \times \ldots \times \mathcal{PD}\mathit{er}\mathcal{C}\mathit{txt}_{l_k}(\mathcal{S}) &\to \mathcal{PD}\mathit{er}\mathcal{C}\mathit{txt}_l(\mathcal{S}) \\ \langle \tilde{\mathcal{PC}}_1, \ldots, \tilde{\mathcal{PC}}_k \rangle &\mapsto \mathcal{PC}[\tilde{\mathcal{PC}}_1, \ldots, \tilde{\mathcal{PC}}_k] \end{aligned} \qquad \text{(B.30)}$$

by structural induction on $\mathcal{PC}$. For this purpose we let $k \in \omega$ and $l_1, \ldots, l_k \in \omega$ be arbitrary, we let $l =_{\text{def}} \max_{1 \le i \le k} l_i$, and then we give, for all pseudo-derivation contexts $\tilde{\mathcal{PC}}_i \in \mathcal{PD}\mathit{er}\mathcal{C}\mathit{txt}_{l_i}(\mathcal{S})$ with $1 \le i \le k$, the following defining clauses (i)′–(iv)′ for the result $\mathcal{PC}[\tilde{\mathcal{PC}}_1, \ldots, \tilde{\mathcal{PC}}_k]$ of *hole-filling* of $\tilde{\mathcal{PC}}_1, \ldots, \tilde{\mathcal{PC}}_k$ into the context-holes $[\,]_1, \ldots, [\,]_k$ occurring in $\mathcal{PC}$; these four clauses are formed in accordance with the defining clauses (i)–(iv) in Definition B.2.13:

(i)′ If $\mathcal{PC}$ is $[\,]_i$, for some $i \in \{1, \ldots, k\}$, then $\mathcal{PC}[\tilde{\mathcal{PC}}_1, \ldots, \tilde{\mathcal{PC}}_k]$ is defined as $\tilde{\mathcal{PC}}_i$.

(ii)′ If $\mathcal{PC}$ is the marked assumption A^u, for some formula $A \in \mathit{Fo}$ and a marker $u \in \mathit{Mk}$, then $\mathcal{PC}[\tilde{\mathcal{PC}}_1, \ldots, \tilde{\mathcal{PC}}_k]$ is defined to be $\mathcal{PC}$ itself.

(iii)′ If $\mathcal{PC}$ is of the form (B.28), for some $A \in \mathit{Fo}$ and $\boldsymbol{R} \in n\mathcal{R}$, then the hole-filling result $\mathcal{PC}[\tilde{\mathcal{PC}}_1, \ldots, \tilde{\mathcal{PC}}_k]$ is also defined to be $\mathcal{PC}$ itself.

(iv)′ If $\mathcal{PC}$ is a pseudo-derivation that is of the form

$$\frac{\mathcal{PC}_1 \qquad \ldots \qquad \mathcal{PC}_n}{A}\ \mathsf{name}(\boldsymbol{R}), u_1, \ldots, u_m$$

for some $A \in \mathit{Fo}$, $\boldsymbol{R} \in n\mathcal{R}$, $n \in \omega \backslash \{0\}$ and $m \in \omega$, and for some pseudo-derivations $\mathcal{PC}_1, \ldots, \mathcal{PC}_n \in \mathcal{PD}\mathit{er}\mathcal{C}\mathit{txt}_k(\mathcal{S})$ and for some assumption markers $u_1, \ldots, u_m \in \mathit{Mk}$, then $\mathcal{PC}[\tilde{\mathcal{PC}}_1, \ldots, \tilde{\mathcal{PC}}_k]$ is defined as

$$\frac{\mathcal{PC}_1[\tilde{\mathcal{PC}}_1, \ldots, \tilde{\mathcal{PC}}_k] \qquad \ldots \qquad \mathcal{PC}_n[\tilde{\mathcal{PC}}_1, \ldots, \tilde{\mathcal{PC}}_k]}{A}\ \mathsf{name}(\boldsymbol{R}), u_1, \ldots, u_m$$

(for the construction of this prooftree the hole-filling operation has been applied, using the induction hypothesis of this definition, to the immediate sub-pseudo-derivation-contexts $\mathcal{PC}_1, \ldots, \mathcal{PC}_n$ of $\mathcal{PC}$).

It is easy to verify that in this way the hole-filling function (B.30) is well-defined: the prooftree $\mathcal{PC}[\tilde{\mathcal{PC}}_1, \ldots, \tilde{\mathcal{PC}}_k]$, which is formed according to this inductive definition for some $\mathcal{PC} \in \mathcal{PD}er\mathcal{C}txt_k(\mathcal{S})$ and $\tilde{\mathcal{PC}}_1, \ldots, \tilde{\mathcal{PC}}_k \in \mathcal{PD}er\mathcal{C}txt(\mathcal{S})$ is indeed a pseudo-derivation in $\mathcal{S}$ that contains only holes that occur in one of $\tilde{\mathcal{PC}}_1, \ldots, \tilde{\mathcal{PC}}_k$.

$\boxtimes$

For the definition of rule derivability in an ANDS $\mathcal{S}$, we need to be able state precisely that the operation of hole-filling in a k-ary derivation context $\mathcal{PC}$ of an ANDS $\mathcal{S}$ with pseudo-derivations $\tilde{\mathcal{PD}}_1, \ldots, \tilde{\mathcal{PD}}_k$ makes the inference steps within $\mathcal{PC}$ "correct". In other words, we want a rigidly defined expression that allows us to formulate, for a k-ary pseudo-derivation context $\mathcal{PC}$ and for pseudo-derivations $\mathcal{PD}_1, \ldots, \mathcal{PD}_k$ in an ANDS $\mathcal{S}$, that in the result $\mathcal{PC}[\tilde{\mathcal{PD}}_1, \ldots, \tilde{\mathcal{PD}}_k]$ of hole-filling the part corresponding to $\mathcal{PC}$ is a correct 'derivation-end' in $\mathcal{S}$: by this we mean more precisely that each inference figure in $\mathcal{PC}[\tilde{\mathcal{PD}}_1, \ldots, \tilde{\mathcal{PD}}_k]$ with rule label $\mathsf{rule}(\boldsymbol{R})$, for some $\boldsymbol{R} \in n\mathcal{R}_{\mathcal{S}}$, which originates from an inference figure in $\mathcal{PC}$ is a correct inference step according to an instance of $\boldsymbol{R}$. For this purpose we introduce, in the definition below, the notion of "derivation context" in an ANDS, which will allow us to express the described situation by saying that a k-ary pseudo-derivation context $\mathcal{PC}$ is a "derivation context" with respect to a sequence $\langle \mathcal{PD}_1, \ldots, \mathcal{PD}_k \rangle$ of pseudo-derivations in $\mathcal{S}$.

Definition B.2.18 (Derivation contexts in ANDS's). Let $\mathcal{S} = \langle Fo, Mk, Na, \mathcal{R} \rangle$ be an ANDS. In item (I) below we define a notion of "derivation context in $\mathcal{S}$ relative to a sequence of pseudo-derivations in $\mathcal{S}$". And in item (II), we stipulate a not relativized version of this notion that does not explicitly refer to a sequence of pseudo-derivations.

(I) Let $k \in \omega$, and let $\tilde{\mathcal{PD}}_1, \ldots, \tilde{\mathcal{PD}}_k \in \mathcal{PD}er(\mathcal{S})$ be pseudo-derivations.

In the inductive clauses (i)″–(iv)″ below, we define, for all pseudo-derivations contexts $\mathcal{PC} \in \mathcal{PD}er\mathcal{C}txt_k(\mathcal{S})$, the property that "$\mathcal{PC}$ *is a* (k-ary) *derivation context in* $\mathcal{S}$ *with respect to* (the sequence) $\langle \tilde{\mathcal{PD}}_1, \ldots, \tilde{\mathcal{PD}}_k \rangle$ (of pseudo-derivations)" by induction on the generation process of pseudo-derivation contexts according to the clauses (i)–(iv) in Definition B.2.13.

(i)″ If $\mathcal{PC}$ is $[\,]_i$, for some $i \in \{1, \ldots, k\}$, then $\mathcal{PC}$ is a derivation context with respect to $\langle \tilde{\mathcal{PD}}_1, \ldots, \tilde{\mathcal{PD}}_k \rangle$.

(ii)″ If $\mathcal{PC}$ is a marked formula A^u, for all $A \in Fo$ and $u \in Mk$, then $\mathcal{PC}$ is a derivation context with respect to $\langle \tilde{\mathcal{PD}}_1, \ldots, \tilde{\mathcal{PD}}_k \rangle$.

(iii)″ If $\mathcal{PC}$ is of the form (B.28), for some $A \in Fo$ and $\boldsymbol{R} \in n\mathcal{R}$, then $\mathcal{PC}$ is a derivation context with respect to $\langle \tilde{\mathcal{PD}}_1, \ldots, \tilde{\mathcal{PD}}_k \rangle$ if and only $\mathcal{PC}$ is a derivation in $\mathcal{S}$ that results from the empty sequence of derivations in $\mathcal{S}$ by an application of R.

(iv)″ If $\mathcal{PC}$ is of the form (B.29), for some named rule $\boldsymbol{R} \in n\mathcal{R}$, a natural number $n \in \omega\backslash\{0\}$, k-ary pseudo-derivation contexts $\mathcal{PC}_1, \ldots, \mathcal{PC}_n$ in $\mathcal{S}$, for some formula $A \in Fo$ and for some assumption markers $u_1, \ldots, u_m \in Mk$, then $\mathcal{PC}$ is a derivation context with respect to $\langle \tilde{\mathcal{PD}}_1, \ldots, \tilde{\mathcal{PD}}_k \rangle$ if and only if the pseudo-derivation $\mathcal{PC}[\tilde{\mathcal{PD}}_1, \ldots, \tilde{\mathcal{PD}}_k]$ of the form

$$\frac{\mathcal{PC}_1[\tilde{\mathcal{PD}}_1, \ldots, \tilde{\mathcal{PD}}_k] \quad \ldots \quad \mathcal{PC}_n[\tilde{\mathcal{PD}}_1, \ldots, \tilde{\mathcal{PD}}_k]}{A}\ \mathsf{name}(\boldsymbol{R}),\ u_1, \ldots, u_m$$

results from $\langle \mathcal{PC}_1[\tilde{\mathcal{PD}}_1, \ldots, \tilde{\mathcal{PD}}_k], \ldots, \mathcal{PC}_n[\tilde{\mathcal{PD}}_1, \ldots, \tilde{\mathcal{PD}}_k] \rangle$ by an application of R, and if furthermore each of $\mathcal{PC}_1, \ldots, \mathcal{PC}_n$ is a derivation context in $\mathcal{S}$ with respect to $\langle \tilde{\mathcal{PD}}_1, \ldots, \tilde{\mathcal{PD}}_k \rangle$.

(II) Let $k \in \omega$, and let $\mathcal{PC} \in \mathcal{PDerCtxt}_k(\mathcal{S})$, i.e. let $\mathcal{PC}$ be a k-ary pseudo-derivation context in $\mathcal{S}$. We say that $\mathcal{PC}$ is a k-ary *derivation context in* $\mathcal{S}$ if and only if there exist pseudo-derivations $\tilde{\mathcal{PD}}_1, \ldots, \tilde{\mathcal{PD}}_k \in \mathcal{PDer}(\mathcal{S})$ such that $\mathcal{PC}$ is a (k-ary) a derivation context in $\mathcal{S}$ with respect to $\langle \tilde{\mathcal{PD}}_1, \ldots, \tilde{\mathcal{PD}}_k \rangle$. We designate by $\mathcal{DerCtxt}_k(\mathcal{S})$ the set of all k-ary derivation contexts in $\mathcal{S}$, and by $\mathcal{DerCtxt}(\mathcal{S})$ the set of all derivation contexts in $\mathcal{S}$, i.e. the union, over all $k \in \omega$, of the sets $\mathcal{DerCtxt}_k(\mathcal{S})$. The concatenated symbols $\mathcal{DC}$ will be used as syntactical variables for derivation contexts.

⊠

As the last prerequisite for the definitions of notions of rule derivability and admissibility in ANDS's, we introduce a "mimicking relation" between derivations in an ANDS. We introduce a 'strict' form of mimicking relation that relates two derivations if and only if they have the same conclusion and the same sets of open assumptions.

Definition B.2.19 (Mimicking (pseudo-)derivations in ANDS's). Let $\mathcal{S}_1$ and $\mathcal{S}_2$ be ANDS's, and let $\mathcal{D}_1$ and $\mathcal{D}_2$ be (pseudo-)derivations in $\mathcal{S}_1$ and in $\mathcal{S}_2$, respectively.

We say that $\mathcal{D}_1$ *mimics* $\mathcal{D}_2$ (or that $\mathcal{D}_1$ is a *mimicking derivation for* $\mathcal{D}_2$, or that $\mathcal{D}_2$ *is mimicked by* $\mathcal{D}_1$), which assertion we denote by $\mathcal{D}_1 \simeq \mathcal{D}_2$, if and only if

$$\mathcal{D}_1 \simeq \mathcal{D}_2 \iff_{\mathrm{def}} \mathsf{concl}(\mathcal{D}_1) = \mathsf{concl}(\mathcal{D}_2)\ \&\ \mathsf{omassm}(\mathcal{D}_1) = \mathsf{omassm}(\mathcal{D}_2) \quad \text{(B.31)}$$

holds, i.e. iff $\mathcal{D}_1$ and $\mathcal{D}_2$ have the same conclusion and the same set of open marked assumptions. ⊠

Lemma B.2.20. *Let $\mathcal{S}$ be an ANDS. Let $k \in \omega$, and let $\mathcal{DC} \in \mathcal{PDerCtxt}_k(\mathcal{S})$ be a derivation context in $\mathcal{S}$ with respect to a sequence $\langle \mathcal{PD}_1, \ldots, \mathcal{PD}_n \rangle$ of pseudo-derivations in $\mathcal{S}$. And furthermore let $\mathcal{D}_1, \ldots, \mathcal{D}_k$ be derivations in $\mathcal{S}$ that mimic $\mathcal{PD}_1, \ldots, \mathcal{PD}_k$, respectively.*

Then $\mathcal{DC}[\mathcal{D}_1, \ldots, \mathcal{D}_k]$ is a derivation in $\mathcal{S}$, which furthermore mimics the pseudo-derivation $\mathcal{DC}[\mathcal{PD}_1, \ldots, \mathcal{PD}_k]$.

At last we are now in a position to give such definitions for the notions of rule derivability, “rule cr-correctness”, “rule cr-admissibility”, and rule admissibility, for which we will then see that for them much carries over from what is known about the relationships between rule derivability and admissibility in pure Hilbert systems (as in particular in APHS’s).

Definition B.2.21 (Rule (cr-)admissibility, cr-correctness, and derivability in an ANDS). Let $\mathcal{S}$ be an ANDS, and let R be an ANDS-rule on $Fo_{\mathcal{S}}$ and $Mk_{\mathcal{S}}$ such that $R = \langle Insts, \mathsf{prem}, \mathsf{concl}, \mathsf{pmassm}, \mathsf{dmassm}\rangle$. And let furthermore $\mathcal{S}^{(+R)}$ be an arbitrary extension of $\mathcal{S}$ by adding R as a new rule.

(i) The rule R is *admissible in* $\mathcal{S}$ (R is an *admissible rule of* $\mathcal{S}$) if and only if

$$\mathcal{S}^{(+R)} \sim_{th} \mathcal{S} \tag{B.32}$$

holds, i.e. iff extending $\mathcal{S}$ by the new rule R with the result $\mathcal{S}^{(+R)}$ does not lead to more theorems in $\mathcal{S}^{(+R)}$.

(ii) The rule R is *cr-correct in* $\mathcal{S}$ (R is a *cr-correct rule for* $\mathcal{S}$, or R is a *correct rule for* $\mathcal{S}$ *with respect to the* $\underline{c}$*onsequence* $\underline{r}$*elation* $\vdash_{\mathcal{S}}$) if and only if

$$(\forall \iota \in Insts)\Big[(\forall i \in \{1,\ldots,\mathsf{arity}(\iota)\})\ \big[\,\mathsf{pmassm}^{(i)}(\iota) \vdash_{\mathcal{S}} \mathsf{prem}^{(i)}(\iota)\,\big] \implies$$

$$\implies \Big(\big(\bigcup_{i=1}^{\mathsf{arity}(\iota)} \mathsf{pmassm}^{(i)}(\iota)\,\big) \setminus \mathsf{dmassm}(\iota)\Big) \vdash_{\mathcal{S}} \mathsf{concl}(\iota)\Big] \tag{B.33}$$

holds, which condition can be paraphrased by saying that the consequence relation $\vdash_{\mathcal{S}}$ on $\mathcal{S}$ is invariant under applications of instances of R.

(iii) The rule R is *cr-admissible in* $\mathcal{S}$ (R is a *cr-admissible rule of* $\mathcal{S}$, or R is *admissible with respect to the* $\underline{c}$*onsequence* $\underline{r}$*elation* $\vdash_{\mathcal{S}}$) if and only if

$$\vdash_{\mathcal{S}^{(+R)}} \; = \; \vdash_{\mathcal{S}} \tag{B.34}$$

holds, i.e. iff the consequence relations $\vdash_{\mathcal{S}^{(+R)}}$ on $\mathcal{S}^{(+R)}$ and $\vdash_{\mathcal{S}}$ on $\mathcal{S}$ (both defined according to Definition B.2.8) coincide, that is more explicitly, iff

$$(\forall \tilde{\Sigma} \in \mathcal{P}_{\mathrm{f}}(mFo_{\mathcal{S}}))(\forall A \in Fo_{\mathcal{S}})\big[\ \tilde{\Sigma} \vdash_{\mathcal{S}^{(+R)}} A \iff \tilde{\Sigma} \vdash_{\mathcal{S}} A\ \big]$$

holds.

(iv) The rule R is *derivable in* $\mathcal{S}$ (R is a *derivable rule of* $\mathcal{S}$) if and only if

$$\begin{aligned}
&(\forall \iota \in Insts_R)\,(\forall n \in \omega,\, n = \mathsf{arity}(\iota))\\
&\quad(\exists\, \mathcal{DC}' \in \mathcal{D}er\mathcal{C}txt_n(\mathcal{S}))\\
&\qquad\big(\forall\, \mathcal{PD} \in \mathcal{D}er(\mathcal{S}^{(+R)})\big)\,\big(\forall\, \mathcal{PD}_1,\ldots,\mathcal{PD}_n \in \mathcal{PD}er(\mathcal{S})\big)\\
&\qquad\quad\Big[\,\mathcal{PD} \text{ results from } \langle \mathcal{PD}_1,\ldots,\mathcal{PD}_n\rangle \text{ by applying } \iota\\
&\qquad\qquad\implies \mathcal{DC}' \text{ is a derivation context w.r.t. } \langle \mathcal{PD}_1,\ldots,\mathcal{PD}_n\rangle\\
&\qquad\qquad\qquad \&\ \mathcal{DC}'[\mathcal{PD}_1,\ldots,\mathcal{PD}_n] \simeq \mathcal{PD}\,\Big]\,.
\end{aligned} \tag{B.35}$$

holds, i.e. if and only if every pseudo-derivation $\mathcal{PD}$ that results from a finite sequence $\langle \mathcal{PD}_1, \ldots, \mathcal{PD}_n \rangle$ of pseudo-derivations in $\mathcal{S}$ by applying an instance of R can be mimicked by a pseudo-derivation in $\mathcal{S}$ of the form $\mathcal{DC}'[\mathcal{PD}_1, \ldots, \mathcal{PD}_n]$ where $\mathcal{DC}'$ is an n-ary derivation-context in $\mathcal{S}$ with respect to $\langle \mathcal{PD}_1, \ldots, \mathcal{PD}_n \rangle$. For an illustration of this defining clause see Figure B.4.

⊠

Proposition B.2.22. *Let $\mathcal{S}$ be an ANDS; let R be an ANDS-rule on $Fo_{\mathcal{S}}$, $Mk_{\mathcal{S}}$.*

The stipulations for "R is admissible in $\mathcal{S}$", "R is cr-admissible in $\mathcal{S}$", and "R is derivable in $\mathcal{S}$" in Definition B.2.21 do not depend on the particular choice, underlying these clauses, of an extension $\mathcal{S}^{(+R)}$ of $\mathcal{S}$ by adding the new rule R (this holds trivially for the stipulation of "R is correct for $\mathcal{S}$" because such extensions do not figure there).

Proof. For the stipulation of rule derivability this is easy to see: only one application of a named version $\boldsymbol{R}$ of the rule R comes into play in this clause, the application of R at the bottom of $\mathcal{PD}$, and the name label of this application does not matter for the condition formulated by this clause.

For the stipulations of rule admissibility and rule cr-admissibility, the statement of the proposition follows from Proposition B.2.12. □

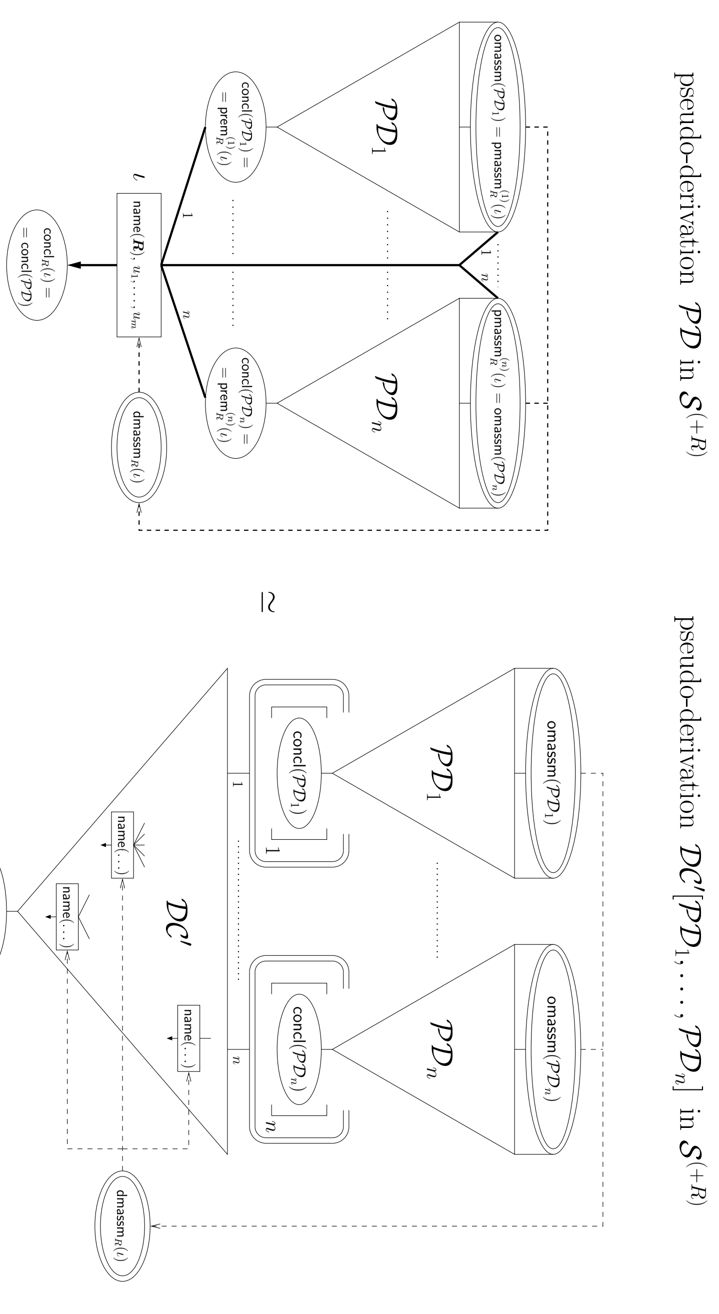
Remark B.2.23. (a) There exists a similarity between the definitions of rule cr-correctness and rule derivability in ANDS's that is not immediately apparent from the defining clauses (B.33) and (B.35). It only becomes evident if the definition of rule cr-correctness is reformulated appropriately.

In order to do so, we let $\mathcal{S}$, R, and $\mathcal{S}^{(+R)}$ be as assumed in Definition B.2.21. Then by expanding (B.33) using the definition of the consequence relation $\vdash_{\mathcal{S}}$ of $\mathcal{S}$ in (B.23) and the stipulation for "$\mathcal{D}$ results from the sequence $\langle \ldots \rangle$ of derivations in $\mathcal{S}$ by an application ι of R" in Definition B.2.5 it follows easily: R is cr-correct for $\mathcal{S}$ if and only if every derivation in $\mathcal{S}^{(+R)}$ that results from a finite sequence of derivations in $\mathcal{S}$ by applying an instance of R can be mimicked by a derivation in $\mathcal{S}$. Or more formally, R is cr-correct for $\mathcal{S}$ iff the following holds:

$$\begin{aligned}
&(\forall \iota \in \mathit{Insts}_R)\,(\forall n \in \omega,\, n = \mathsf{arity}(\iota))\\
&\quad(\exists\, \mathcal{D}' \in \mathcal{D}er(\mathcal{S}))\\
&\qquad(\forall\, \mathcal{D} \in \mathcal{D}er(\mathcal{S}^{(+R)}))\,(\forall\, \mathcal{D}_1, \ldots, \mathcal{D}_n \in \mathcal{D}er(\mathcal{S}))\\
&\qquad\quad\big[\, \mathcal{D} \text{ results from } \langle \mathcal{D}_1, \ldots, \mathcal{D}_n \rangle \text{ by an applying } \iota \;\Longrightarrow\\
&\qquad\qquad\qquad\qquad\Longrightarrow\; \mathcal{D}' \simeq \mathcal{D} \,\big]\,. \qquad \text{(B.36)}
\end{aligned}$$

(For an illustration of this condition see Figure B.5). Assuming that the rule R is indeed of the form $R = \langle \mathit{Insts}, \mathsf{prem}, \mathsf{concl}, \mathsf{pmassm}, \mathsf{dmassm} \rangle$, as presupposed in Definition B.2.21, it hence is the case that the clauses (B.33) and

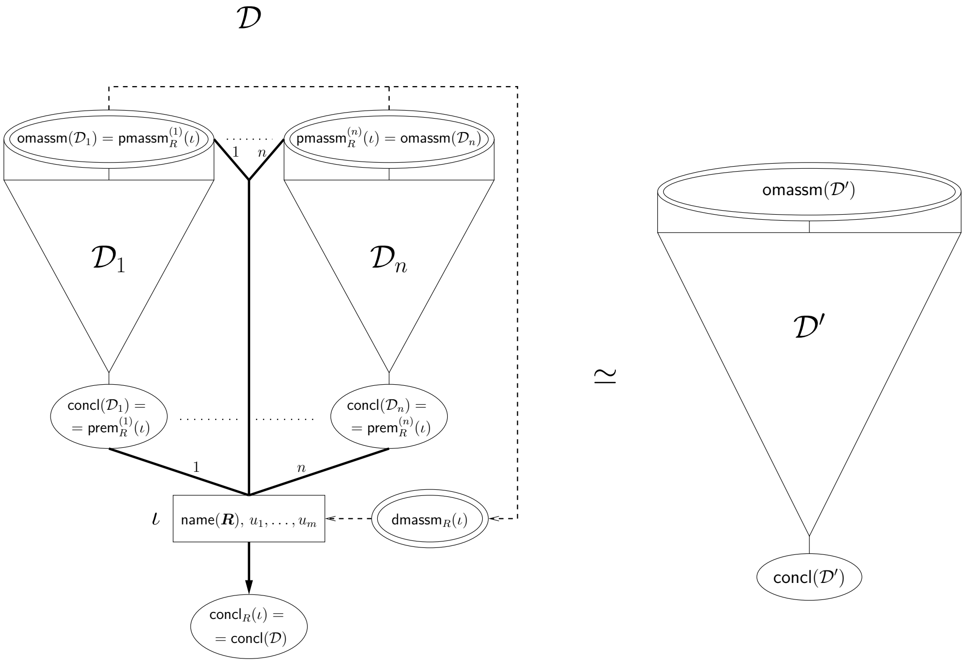
Figure B.4: Illustration of the defining clause (B.35) for derivability of an ANDS-rule R with respect to an ANDS $\mathcal{S}$: "mimicking" R-applications by appropriate derivation contexts in $\mathcal{S}$.


pseudo-derivation $\mathcal{PD}$ in $\mathcal{S}^{(+R)}$
omassm($\mathcal{PD}_1$) = pmassm$_R^{(1)}(\iota)$
pmassm$_R^{(n)}(\iota)$ = omassm($\mathcal{PD}_n$)
1
n
$\mathcal{PD}_1$
$\mathcal{PD}_n$
concl($\mathcal{PD}_1$) = = prem$_R^{(1)}(\iota)$
concl($\mathcal{PD}_n$) = = prem$_R^{(n)}(\iota)$
1
n
ι
name($\boldsymbol{R}$), $u_1, \ldots, u_m$
dmassm$_R(\iota)$
concl$_R(\iota)$ = = concl($\mathcal{PD}$)
$\simeq$
pseudo-derivation $\mathcal{DC}'[\mathcal{PD}_1, \ldots, \mathcal{PD}_n]$ in $\mathcal{S}^{(+R)}$
omassm($\mathcal{PD}_1$)
omassm($\mathcal{PD}_n$)
$\mathcal{PD}_1$
$\mathcal{PD}_n$
concl($\mathcal{PD}_1$)
1
concl($\mathcal{PD}_n$)
n
1
n
$\mathcal{DC}'$
name(...)
name(...)
name(...)
dmassm$_R(\iota)$
concl($\mathcal{DC}'$)

$\mathcal{PD}$ results from the sequence
$\langle \mathcal{PD}_1, \ldots, \mathcal{PD}_n \rangle$ of pseudo-derivations in $\mathcal{S}$
by an application of the instance ι of R
($\boldsymbol{R}$ is the named version of R added to $\mathcal{S}$ in $\mathcal{S}^{(+R)}$)

$\mathcal{DC}'$ is an n-ary pseudo-derivation-context in $\mathcal{S}$
that is also a derivation-context in $\mathcal{S}$
with respect to $\langle \mathcal{PD}_1, \ldots, \mathcal{PD}_n \rangle$

Figure B.5: Illustration of the alternative defining clause (B.36) for cr-correctness of an ANDS-rule R with respect to an ANDS $\mathcal{S}$: for every derivation $\mathcal{D}$ in a system $\mathcal{S}^{(+R)}$ (which is an extension of $\mathcal{S}$ by adding a new named version $\boldsymbol{R}$ of the rule R) that results from a sequence $\langle \mathcal{D}_1, \ldots, \mathcal{D}_n \rangle$ of derivations in $\mathcal{S}$ by an application of an instance ι of R, there exists a mimicking derivation $\mathcal{D}'$ in $\mathcal{S}$.


$\mathcal{D}$
omassm($\mathcal{D}_1$) = pmassm$^{(1)}_R(\iota)$
pmassm$^{(n)}_R(\iota)$ = omassm($\mathcal{D}_n$)
1 n
$\mathcal{D}_1$
$\mathcal{D}_n$
concl($\mathcal{D}_1$) = = prem$^{(1)}_R(\iota)$
concl($\mathcal{D}_n$) = = prem$^{(n)}_R(\iota)$
1
n
ι
name($\boldsymbol{R}$), $u_1, \ldots, u_m$
dmassm$_R(\iota)$
concl$_R(\iota)$ = = concl($\mathcal{D}$)
$\simeq$
omassm($\mathcal{D}'$)
$\mathcal{D}'$
concl($\mathcal{D}'$)

(B.36) are equivalent; this holds irrespective of which particular extension $\mathcal{S}^{(+R)}$ of $\mathcal{S}$ by adding the new rule R has actually been chosen for the clause (B.36) (this is easy to see, by an analogous argumentation as the one used for rule derivability in the proof of Proposition B.2.22).

Now, (B.36) and (B.35) are clearly statements of a similar form. And furthermore, it follows easily from Proposition B.2.22 that (B.35) implies (B.36). That is, derivability of a rule R in an ANDS $\mathcal{S}$ implies cr-correctness of R for $\mathcal{S}$. (In this way we have proved the first sentence of Lemma B.2.24, (ii), below.)

(b) Because the defining clauses (B.33) and (B.35) for rule cr-correctness and rule derivability (as well as the alternative one (B.36) for rule cr-correctness) in ANDS's are not yet quite illustrative, we want to expand them to reach more explicit 'practical usable formulations' of these clauses. For this, we let $\mathcal{S}$, R, and $\mathcal{S}^{(+R)}$ be given arbitrarily such that the assumption of Definition B.2.21

is fulfilled; we assume that the named version of R that is added to $\mathcal{S}$ in $\mathcal{S}^{(+R)}$ is the named rule $\boldsymbol{R}$.

For rule cr-correctness we find that R is cr-correct in $\mathcal{S}$ if and only if the following two conditions are fulfilled:

- For every application $\iota \in \mathit{Insts}_R$ with $\mathsf{arity}(\iota) = 0$ there exists a derivation $\mathcal{D}' \in \mathcal{D}er(\mathcal{S})$ with conclusion $\mathsf{concl}(\iota)$ and without open assumptions.
- For every application $\iota \in \mathit{Insts}_R$ with $\mathsf{arity}(\iota) = n \geq 1$, and for every derivation $\mathcal{D} \in \mathcal{D}er(\mathcal{S}^{(+R)})$ of the form

$$\frac{\begin{array}{ccc}\mathcal{D}_1 & & \mathcal{D}_n \\ \mathsf{prem}_R^{(1)}(\iota) & \dots & \mathsf{prem}_R^{(n)}(\iota)\end{array}}{\mathsf{concl}_R(\iota)}\ \mathsf{name}(\boldsymbol{R}),\ u_1,\dots,u_m \tag{B.37}$$

  that results from the sequence $\langle \mathcal{D}_1,\dots,\mathcal{D}_n\rangle$ of derivations in $\mathcal{S}$ by an application of ι there exists a derivation $\mathcal{D}' \in \mathcal{D}er(\mathcal{S})$ of the form

$$\begin{array}{c}\mathcal{D}' \\ \mathsf{concl}_R(\iota)\end{array}$$

  such that $\mathsf{omassm}(\mathcal{D}') = \mathsf{omassm}(\mathcal{D})$.

And concerning rule derivability we find that R is derivable in $\mathcal{S}$ if and only if the following two conditions are satisfied:

- For every application $\iota \in \mathit{Insts}_R$ with $\mathsf{arity}(\iota) = 0$ there exists a derivation $\mathcal{D}' \in \mathcal{D}er(\mathcal{S})$ with conclusion $\mathsf{concl}(\iota)$ and without open assumptions.
- For every application $\iota \in \mathit{Insts}_R$ with $\mathsf{arity}(\iota) = n \geq 1$, and for every pseudo-derivation $\mathcal{PD} \in \mathcal{D}er(\mathcal{S}^{(+R)})$

$$\frac{\begin{array}{ccc}\mathcal{PD}_1 & & \mathcal{PD}_n \\ \mathsf{prem}_R^{(1)}(\iota) & \dots & \mathsf{prem}_R^{(n)}(\iota)\end{array}}{\mathsf{concl}_R(\iota)}\ \mathsf{name}(\boldsymbol{R}),\ u_1,\dots,u_m$$

  that results from the sequence $\langle \mathcal{PD}_1,\dots,\mathcal{PD}_n\rangle$ of pseudo-derivations in $\mathcal{S}$ by an application of ι there exists a pseudo-derivation $\mathcal{PD}' \in \mathcal{D}er(\mathcal{S})$ of the form

$$\begin{array}{ccc}\mathcal{PD}_1 & & \mathcal{PD}_n \\ [\mathsf{prem}_R^{(1)}(\iota)]_1 & \dots & [\mathsf{prem}_R^{(n)}(\iota)]_n \\ & \mathcal{DC} & \\ & \mathsf{concl}_R(\iota) & \end{array}$$

where $\mathcal{DC}$ is an n-ary derivation context in $\mathcal{S}$ with respect to the sequence $\langle \mathcal{PD}_1, \ldots, \mathcal{PD}_n \rangle$ such that $\mathsf{omassm}(\mathcal{PD}') = \mathsf{omassm}(\mathcal{PD})$ holds, i.e. such that $\mathcal{PD}$ and $\mathcal{PD}'$ have the same open assumptions (and hence $\mathcal{PD}$ and $\mathcal{PD}'$ mimic each other, since they also have the same conclusion).

Now the following lemma is a counterpart in ANDS's of Lemma B.1.11 with respect to the four notions "admissible", "cr-correct", "cr-admissible", and "derivable" concerning ANDS-rules. Rule derivability turns out to be similarly behaved as in APHS's. Rule admissibility is a weaker notion than any of "derivability", "cr-correctness" and "cr-admissibility"; and the latter two notions coincide.

The following lemma contains the useful statement that hole-filling in derivation-contexts of an ANDS $\mathcal{S}$ with derivations of $\mathcal{S}$ results in a derivation in $\mathcal{S}$.

Lemma B.2.24. *Let $\mathcal{S}$ be an ANDS and let R be an ANDS-rule on $Fo_{\mathcal{S}}$ and $Mk_{\mathcal{S}}$. Then the following statements holds:*

(i) R is cr-correct for $\mathcal{S}$ $\iff$ R is cr-admissible in $\mathcal{S}$.

(ii) If R is derivable in $\mathcal{S}$, then R is also cr-admissible in $\mathcal{S}$. If R is cr-admissible in $\mathcal{S}$, then R is also admissible in $\mathcal{S}$. Neither implication in the opposite direction holds in general.

(iii) If R is derivable in $\mathcal{S}$, then R is derivable in every extension by enlargement of $\mathcal{S}$.

Eventually we turn to the question of what precise consequences the respective property of a rule R to be either admissible, cr-admissible, or derivable in an ANDS $\mathcal{S}$ has for the possibility to eliminate applications of a named version $\boldsymbol{R}$ of R from derivations in an extension $\mathcal{S}^{(+R)}$ of $\mathcal{S}$ by adding the new rule R (in the form of the named rule $\boldsymbol{R}$). For this purpose, we define now three abstract notions of "rule elimination" in an ANDS $\mathcal{S}$, which respectively formulate the conditions that applications of a named rule $\boldsymbol{R}$ of $\mathcal{S}$ "can be eliminated" (i) from arbitrary derivations in $\mathcal{S}$ without open assumptions, (ii) from arbitrary derivations in $\mathcal{S}$, or (iii) from arbitrary derivation contexts in $\mathcal{S}$; hereby the expression "can be eliminated" is to be understood as the assertion of the existence of respective mimicking derivations (that, in case (iii), must be of a certain form).

For the sake of convenience in the formulation of the definition below (as well as of the subsequent theorem), we reformulate this question as the problem of determining, for arbitrary ANDS's $\mathcal{S}$ and named rules $\boldsymbol{R}$ of $\mathcal{S}$, the respective consequences that admissibility, cr-admissibility and derivability of $\mathsf{rule}(\boldsymbol{R})$ in the ANDS $\mathcal{S}-\boldsymbol{R}$ have for the possibility of eliminating applications of $\boldsymbol{R}$ from derivations in $\mathcal{S}$; hereby $\mathcal{S}-\boldsymbol{R}$ stands for the ANDS $\langle Fo, Mk, Na, n\mathcal{R}\backslash\{\boldsymbol{R}\}\rangle$ that results by removing a named rule $\boldsymbol{R} \in n\mathcal{R}$ from the set of rules of the ANDS $\mathcal{H} = \langle Fo, Mk, Na, n\mathcal{R}\rangle$ (we will use this notation also in similar situations).

Definition B.2.25 (Three notions of rule elimination). Let $\mathcal{S}$ be an ANDS, and let $\boldsymbol{R}$ be a named rule of $\mathcal{S}$. In the three items below, we respectively define three notions of "rule elimination".

(i) We say that $\boldsymbol{R}$-*elimination holds in* $\mathcal{S}$ (or that $\boldsymbol{R}$-*elimination holds in* $\mathcal{D}er_{\emptyset}(\mathcal{S})$) if and only if

$$(\forall \mathcal{D} \in \mathcal{D}er_{\emptyset}(\mathcal{S}))\,(\exists\, \mathcal{D}' \in \mathcal{D}er(\mathcal{S}{-}\boldsymbol{R}))\,\big[\,\mathcal{D}' \simeq \mathcal{D}\,\big] \tag{B.38}$$

holds, i.e. iff every derivation in $\mathcal{S}$ without open assumptions can be mimicked by a derivation in $\mathcal{S}$ that does not contain applications of $\boldsymbol{R}$.

(ii) We stipulate that $\boldsymbol{R}$-*elimination holds for* $\mathcal{D}er(\mathcal{S})$ if and only if

$$(\forall \mathcal{D} \in \mathcal{D}er(\mathcal{S}))\,(\exists\, \mathcal{D}' \in \mathcal{D}er(\mathcal{S}{-}\boldsymbol{R}))\,\big[\,\mathcal{D}' \simeq \mathcal{D}\,\big] \tag{B.39}$$

holds, i.e. iff for every derivation in $\mathcal{S}$ there exists a mimicking derivation in $\mathcal{S}$ without applications of $\boldsymbol{R}$.

(iii) We define that $\boldsymbol{R}$-*elimination holds for* $\mathcal{D}er\mathcal{C}txt(\mathcal{S})$ if and only if

$$\begin{aligned}
&(\forall n \in \omega)\,(\forall\, \mathcal{DC} \in \mathcal{PD}er\mathcal{C}txt_n(\mathcal{S}))\,\big(\forall\, \mathcal{PD}_1,\ldots,\mathcal{PD}_n \in \mathcal{PD}er(\mathcal{S})\big)\\
&\quad(\exists\, \mathcal{DC}' \in \mathcal{PD}er\mathcal{C}txt_n(\mathcal{S}{-}\boldsymbol{R}))\\
&\qquad\Big[\,\mathcal{DC} \text{ is derivation context w.r.t. } \langle \mathcal{PD}_1,\ldots,\mathcal{PD}_n\rangle\\
&\qquad\quad\Longrightarrow\;\; \mathcal{DC}' \text{ is a derivation context w.r.t. } \langle \mathcal{PD}_1,\ldots,\mathcal{PD}_n\rangle\\
&\qquad\qquad\quad \&\;\; \mathcal{DC}'[\mathcal{PD}_1,\ldots,\mathcal{PD}_n] \;\simeq\; \mathcal{DC}[\mathcal{PD}_1,\ldots,\mathcal{PD}_n]\,\Big]
\end{aligned} \tag{B.40}$$

holds, i.e. iff, for every pseudo-derivation context $\mathcal{DC}$ in $\mathcal{S}$ that is a derivation context with respect to the sequence $\langle \mathcal{PD}_1,\ldots,\mathcal{PD}_n\rangle$ of pseudo-derivations in $\mathcal{S}$, the applications of $\boldsymbol{R}$ can be eliminated with the result of a derivation context $\mathcal{DC}'$ with respect to $\langle \mathcal{PD}_1,\ldots,\mathcal{PD}_n\rangle$ in $\mathcal{S}{-}\boldsymbol{R}$ such that the respective results of hole-filling with $\mathcal{PD}_1,\ldots,\mathcal{PD}_n$ in $\mathcal{DC}$ and in $\mathcal{DC}'$ are mimicking pseudo-derivations of each other. ⊠

The notions of rule elimination defined in items (i) and (ii) of the definition above are clearly natural choices for such notions, whereas this is, perhaps, less obvious for the notion in item (iii). However, it is easy to see the similarity of the formal condition (B.40) with the defining clause (B.35) for rule derivability in an ANDS. The relevance of this third notion is furthermore underscored by the following theorem, which characterizes the three notions of rule elimination in terms of a respective one from the notions of rule admissibility, cr-admissibility, and derivability.

Eventually the following theorem establishes respective correspondences between the three notions of rule elimination defined above with the notions of rule admissibility, cr-admissibility, and derivability in ANDS's..

Theorem B.2.26 (Rule elimination vs. rule (cr-)admissibility and derivability). *Let* $\mathcal{S}$ *be an ANDS, and let* $\boldsymbol{R}$ *be a named rule of* $\mathcal{S}$*. The following three statements hold:*

(i) $\boldsymbol{R}$*-elimination holds in* $\mathcal{S}$ $\iff$ $\boldsymbol{R}$ *is admissible in* $\mathcal{S}{-}R\,.$

(ii) $\boldsymbol{R}$*-elimination holds in* $\mathcal{D}er(\mathcal{S})$ $\iff$ $\boldsymbol{R}$ *is cr-admissible in* $\mathcal{S}{-}\boldsymbol{R}\,.$

(iii) $\boldsymbol{R}$*-elimination holds in* $\mathcal{D}er\mathcal{C}txt(\mathcal{S})$ $\iff$ $\boldsymbol{R}$ *is derivable in* $\mathcal{S}{-}\boldsymbol{R}\,.$

Appendix C

Derivations in $\mathbf{HB}_{\mathbf{0}}^{=}$ Without Redundancies

In this appendix we are concerned with the depth of derivations in $\mathbf{HB}_{\mathbf{0}}^{=}$. Our main theorem here is a bounding function on the depth of derivations in $\mathbf{HB}_{\mathbf{0}}^{=}$ that do not contain certain kinds of redundancies. Other results provide conditions under which redundancies of respective kind can effectively be eliminated from $\mathbf{HB}_{\mathbf{0}}^{=}$-derivations. Furthermore, we show two corollaries to the main theorem in this appendix. The first one states that every derivation $\mathcal{D}'$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ without open assumptions can be transformed into a derivation $\mathcal{D}'$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ that mimics $\mathcal{D}$ and for which its depth $|\mathcal{D}'|$ can be bounded using the assertion of the main theorem. And the second corollary states that if a derivation $\mathcal{D}$ without the mentioned kind of redundancies contains open assumptions only above a certain limit-height (which depends on the sizes and syntactical depths of the recursive types in the conclusion of $\mathcal{D}$), then the open assumptions of $\mathcal{D}$ can be eliminated with the result of a derivation $\mathcal{D}'$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ without open assumptions that mimics $\mathcal{D}$.

The results in this appendix are designed to be used as tools for showing termination of procedures that build up derivations in $\mathbf{HB}_{\mathbf{0}}^{=}$ in a stepwise and systematic manner. For instance, they can be used to reconstruct the completeness proof that [BrHe98] have given for the system $\mathbf{HB}^{=}$ with respect to recursive type equality $=_{\mu}$ into a completeness proof with respect to $=_{\mu}$ of the 'analytic' version $\mathbf{HB}_{\mathbf{0}}^{=}$ of $\mathbf{HB}^{=}$ with respect to $=_{\mu}$. In particular, the results here can be applied to show termination of a procedure analogous to one[1] given by [BrHe98] that is able to build up a derivation $\mathcal{D}$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ from an arbitrary given conclusion $\tau = \sigma$ with $\tau, \sigma \in \mu Tp$ such that $\tau =_{\mu} \sigma$.

In the context of the investigations here, results of this appendix are used in two other situations: namely, for showing termination of the procedures in the

[1] We refer to Algorithm S in Figure 5 on p.11 in [BrHe98], which is used there to show completeness of the axiomatization given by Brandt and Henglein of the recursive subtyping relation $\leq_{\mu}$, and we mean a straightforward modification of this algorithm that can be used to show completeness with respect to $=_{\mu}$ of the proof system $\mathbf{HB}^{=}$.

proofs of Lemma 7.2.9 in Section 7.2, and of Theorem 8.2.2 in Section 8.2. In both of these proofs, sequences $\mathcal{SD} = \langle \mathcal{D}^{(n)} \rangle_{n \in I}$ of derivations in $\mathbf{HB}^{=}_{\mathbf{0}}$ of a certain regular and redundancy-free kind, where $I = \omega$ or $I = \{0, 1, \ldots, n_{\max}\}$ for some $n_{\max} \in \omega$, are effectively generated by respective procedures such that the minimal heights h_n of open marked assumptions in $\mathbf{HB}^{=}_{\mathbf{0}}$ are strictly increasing and, if $\mathcal{SD}$ is an infinite sequence, tend to infinity.[2] In reasoning about such sequences $\mathcal{SD}$, respective statements proved in this appendix are then used to show that either $\mathcal{SD}$ cannot be infinite (in the case of the proof for Lemma 7.2.9), or that, if $\mathcal{SD}$ is infinite, then a derivation $\mathcal{D}'$ in $\mathbf{HB}^{=}_{\mathbf{0}}$ with the same conclusion as all derivations in $\mathcal{SD}$ and without open assumptions can effectively be 'extracted' from a derivation $\mathcal{D}^{(n)}$ in $\mathcal{SD}$ with n "sufficiently large" (in the case of the proof of Theorem 8.2.2).

The main theorem of this appendix, Theorem C.11, states the following: every derivation $\mathcal{D}$ in $\mathbf{HB}^{=}_{\mathbf{0}}$ with or without open assumption classes that does not contain certain specific kinds of redundancies is bounded in its depth by an easy function dependent on the sizes and the syntactical depths of the recursive types in the conclusion of $\mathcal{D}$. Hereby the following three kinds of redundancies will be excluded from occurring in a derivation $\mathcal{D}$ to which the bound stated by this theorem applies:

(Rdcy1) the occurrence in $\mathcal{D}$ of two or more successive applications of the bound-variable renaming rule REN,

(Rdcy2) the occurrence in $\mathcal{D}$ of "detours" that are caused by *certain* applications of rules $\mathrm{FOLD}_{l/r}$ for which there exist applications of rules $(\mu\!-\!\bot)^{\bot\mathrm{der}}_{l/r}$ with the same conclusion and which will be called "nlμb-decreasing" applications of $\mathrm{FOLD}_{l/r}$, and

(Rdcy3) the occurrence in $\mathcal{D}$ of "detours" of the sort described in the following remark that arise if the power of the rule ARROW/FIX of $\mathbf{HB}^{=}_{\mathbf{0}}$ to discharge open assumptions is not used thoroughly enough in $\mathcal{D}$.

Redundancies of kind (Rdcy1) in $\mathbf{HB}^{=}_{\mathbf{0}}$ are the most obvious ones: due to the transitivity of the variant relation $\equiv_{\mathrm{ren}}$ on recursive types, successive applications of REN in a $\mathbf{HB}^{=}_{\mathbf{0}}$-derivation $\mathcal{D}$ can always be replaced by a single application of REN without affecting the conclusion or the open assumptions of $\mathcal{D}$. Redundancies of kind (Rdcy2) will be described and treated later (on pages 402–405). We start, however, by explaining redundancies of kind (Rdcy3), which are related to the use of the rule ARROW/FIX in a $\mathbf{HB}^{=}_{\mathbf{0}}$-derivation $\mathcal{D}$ for discharging marked formulas that have been encountered earlier in $\mathcal{D}$.

Remark C.1 (Redundancies of kind (Rdcy3) in $\mathbf{HB}^{=}_{\mathbf{0}}$-derivations). Suppose that a derivation $\mathcal{D}$ in $\mathbf{HB}^{=}_{\mathbf{0}}$ is given that possibly contains open assumption

[2] In the case of the proof of Theorem 8.2.2 such a sequence is only encountered indirectly as the sequence $\langle \lceil \mathcal{D}^{(n)} \rceil^{\{u_i\}_i}_{\mathbf{HB}^{=}_{\mathbf{0}}} \rangle_{n \in I}$ of $\mathbf{HB}^{=}_{\mathbf{0}}$-end-derivations of some sequence $\langle \mathcal{D}^{(n)} \rangle_{n \in I}$ of derivations with the same conclusions in the extension of $\mathbf{HB}^{=}_{\mathbf{0}}$ with SYMM and the generalized transitivity rules $\{\mathrm{TRANS}_k\}_k$.

classes and that is of the form

$$\left.\begin{array}{c}\cfrac{\begin{array}[b]{c}\mathcal{D}_3\\(\tau_1' \to \tau_2' = \sigma_1' \to \sigma_2')\\\mathcal{DC}_1\\\tau_1 = \sigma_1\end{array}\qquad\qquad\begin{array}[b]{c}\mathcal{D}_2\\\tau_2 = \sigma_2\end{array}}{(\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2)}\;\begin{array}{l}\text{ARROW}\\\quad\text{or ARROW/FIX}, u\end{array}\\\mathcal{DC}_0\\\tau = \sigma\end{array}\right\}\tag{C.1}$$

with some $\tau_1, \tau_2, \sigma_1, \sigma_2, \tau_1', \tau_2', \sigma_1', \sigma_2' \in \mu Tp$ such that, for each $i \in \{1, 2\}$, $\tau_i' \equiv_{\text{ren}} \tau_i$ and $\sigma_i' \equiv_{\text{ren}} \sigma_i$ is the case, with derivation contexts $\mathcal{DC}_0$ and $\mathcal{DC}_1$, subderivations $\mathcal{D}_2$ and $\mathcal{D}_3$, and with conclusion $\tau = \sigma$, for some $\tau, \sigma \in \mu Tp$.

In this situation we observe the following: if the subderivation $\mathcal{D}_3$ of $\mathcal{D}$ consists of anything else but of a single marked assumption $(\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2)^u$ that is followed, down to the conclusion $\tau_1' \to \tau_2' = \sigma_1' \to \sigma_2'$ of $\mathcal{D}_3$, by zero, one or more applications of REN, and that is, as a marked assumption in $\mathcal{D}$, discharged at precisely the displayed application of ARROW/FIX (in which case the displayed application can therefore not be an application of ARROW), then the subderivation $\mathcal{D}_3$ of $\mathcal{D}$ can be looked upon as an unnecessary *detour* of $\mathcal{D}$ (we will later also speak of $\mathcal{D}_3$ of a *detour subderivation* of $\mathcal{D}$) This is because in this case the derivation $\mathcal{D}$ can be shortened by replacing $\mathcal{D}_3$ through a single application of REN whose premise is a marked assumption that is discharged in the resulting derivation $\mathcal{D}'$: More precisely, $\mathcal{D}$ can be transformed into the derivation $\mathcal{D}'$ of the simpler form

$$\left.\begin{array}{c}\cfrac{\begin{array}[b]{c}\cfrac{(\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2)^v}{(\tau_1' \to \tau_2' = \sigma_1' \to \sigma_2')}\,\text{REN}\\\mathcal{DC}_1\\\tau_1 = \sigma_1\end{array}\qquad\qquad\begin{array}[b]{c}\mathcal{D}_2^{([v/u])}\\\tau_2 = \sigma_2\end{array}}{(\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2)}\;\text{ARROW/FIX}, v\\\mathcal{DC}_0\\\tau = \sigma\end{array}\right\}\tag{C.2}$$

where v is a fresh assumption marker not present in $\mathcal{D}$, and $\mathcal{D}_2^{([v/u])}$ is either $\mathcal{D}_2$, if the displayed rule application in (C.1) is an application of ARROW, or, if the displayed rule application in (C.1) is an application of ARROW/FIX at which assumptions $(\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2)^u$ are discharged, it results from $\mathcal{D}_2$ by replacing the open marked assumptions $(\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2)^u$ by the open marked assumptions $(\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2)^v$. Hence the marked assumption $(\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2)^v$ in (C.2) is discharged at the displayed application of ARROW/FIX. The derivation $\mathcal{D}'$ has the same conclusion as $\mathcal{D}$ and possibly fewer open assumptions, that is, $\mathcal{D}'$ mimics $\mathcal{D}$.

Obviously, an analogous situation as (C.1) arises if there is an occurrence of the formula $\tau_1' \to \tau_2' = \sigma_1' \to \sigma_2'$ with a similar property at the top of the subderivation

$\mathcal{D}_2$ of $\mathcal{D}$ instead of at the top of the subderivation $\mathcal{DC}_1$ as is the case in the symbolic prooftree (C.1).

With respect to notions introduced in two definitions below, we will say, for a derivation $\mathcal{D}$ of the form (C.1) and with the properties supposed above on the occurring subderivations and recursive types, that the displayed formula occurrence $\tau_1' \to \tau_2' = \sigma_1' \to \sigma_2'$ is *not* "associated with an assumption of $\mathcal{D}$" *nor* "associated with a discharged assumption of $\mathcal{D}$", and that due to this "$\mathcal{D}$ does *not* fulfill the condition **ADA**". And in the special situation that the subderivation $\mathcal{D}_3$ in a derivation $\mathcal{D}$ of the form (C.1) consists only of a sequence of REN-applications with an open marked assumption of $\mathcal{D}$ at its top, we will say that "$\mathcal{D}$ does not fulfill the condition **AA**".

In the following definition we introduce two designations for such formula occurrences in $\mathbf{HB}_0^{=}$-derivations that are not the conclusion of a 'detour' derivation as described in Remark C.1. We stipulate an occurrence of a formula in a $\mathbf{HB}_0^{=}$-derivation to be "associated with an assumption" if and only if there are only REN-applications and a marked assumption above this formula occurrence; and we define in a similar way when a formula occurrence in a $\mathbf{HB}_0^{=}$-derivation is "associated with a discharged assumption".

Definition C.2. (Formula-occurrences in $\mathbf{HB}_0^{=}$-derivations that are associated with assumptions). Let $\mathcal{D}$ be a derivation in $\mathbf{HB}_0^{=}$ with possibly open assumptions. Furthermore, let an occurrence of a formula $\chi_1 = \chi_2$, for some $\chi_1, \chi_2 \in \mu Tp$, in $\mathcal{D}$ be considered.

(i) We say that the considered formula occurrence of $\chi_1 = \chi_2$ in $\mathcal{D}$ *is associated with an assumption of* $\mathcal{D}$ if and only if it is reachable from a marked assumption at the top of $\mathcal{D}$ by a thread in $\mathcal{D}$ which only passes applications of rules REN (in particular this is the case if the considered formula occurrence takes place within a marked assumption of $\mathcal{D}$).

(ii) We say that the considered formula occurrence of $\chi_1 = \chi_2$ in $\mathcal{D}$ *is associated with a discharged assumption of* $\mathcal{D}$ if and only if it is reachable by a thread in $\mathcal{D}$ which passes only applications of rules REN from an assumption of $\mathcal{D}$ at the top that is discharged in $\mathcal{D}$ (again, this is the case in particular if the considered formula occurrence takes place within a marked assumption of $\mathcal{D}$ that is discharged in $\mathcal{D}$).

⊠

Now we define two properties of $\mathbf{HB}_0^{=}$-derivations, fulfilledness of the condition **AA**, and fulfilledness of the condition **ADA**. Hereby condition **ADA** is designed for the purpose of excluding detours in $\mathbf{HB}_0^{=}$-derivations of the kind described in Remark C.1; and condition **AA** is a weakening of condition **ADA**.

Definition C.3 (The conditions AA and ADA for $\mathbf{HB}_0^{=}$-derivations). Let $\mathcal{D}$ be a derivation in $\mathbf{HB}_0^{=}$ that possibly contains open assumption classes.

(i) We say that $\mathcal{D}$ *satisfies*, or *fulfills*, *the condition* **AA** ("association with assumptions") if and only if the following holds: Each formula occurrence o in $\mathcal{D}$ of a formula $\tau_1' \to \tau_2' = \sigma_1' \to \sigma_2'$, for some $\tau_1', \tau_2', \sigma_1', \sigma_2' \in \mu Tp$, for which the assertion

> "On the thread downwards from the formula occurrence o to the conclusion of $\mathcal{D}$, after passing at least one application of a rule that is different from REN, a formula $\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2$ is encountered, where $\tau_1, \tau_2, \sigma_1, \sigma_2 \in \mu Tp$ such that $\tau_i \equiv_{\text{ren}} \tau_i'$ and $\sigma_i \equiv_{\text{ren}} \sigma_i'$ for each $i \in \{1, 2\}$." } (C.3)

holds, is associated with an assumption of $\mathcal{D}$.

(ii) We say that $\mathcal{D}$ *satisfies*, or *fulfills*, *the condition* **ADA** ("association with discharged assumptions") if and only if the following holds: Each formula occurrence in $\mathcal{D}$ of a formula $\tau_1' \to \tau_2' = \sigma_1' \to \sigma_2'$, for some $\tau_1', \tau_2', \sigma_1', \sigma_2' \in \mu Tp$, for which (C.3) holds, is associated with a discharged assumption of $\mathcal{D}$.

⊠

Obviously, the condition **ADA** is stronger than the condition **AA**, i.e. every derivation in $\mathbf{HB}_{\mathbf{0}}^{=}$ that fulfills **ADA** fulfills **AA** as well; and it is easy to give examples showing that the reverse implication is not the case. Furthermore, we observe the following consequence of the fact that the only two rules of $\mathbf{HB}_{\mathbf{0}}^{=}$ apart from the rule REN that have applications with formulas $\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2$ as conclusions, are the rules ARROW and ARROW/FIX: The assertion (C.3) in Definition C.3 can be replaced by the assertion

> "On the thread downwards from the formula occurrence o to the conclusion of $\mathcal{D}$, after passing at least one application of ARROW or ARROW/FIX, a formula $\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2$ is encountered, where $\tau_1, \tau_2, \sigma_1, \sigma_2 \in \mu Tp$ such that $\tau_i \equiv_{\text{ren}} \tau_i'$ and $\sigma_i \equiv_{\text{ren}} \sigma_i'$ for each $i \in \{1, 2\}$." } (C.4)

without changing the respective definitions of the conditions **AA** and **ADA**. This observation implies the following proposition.

Proposition C.4. *Let $\mathcal{D}$ be a derivation in $\mathbf{HB}_{\mathbf{0}}^{=}$ that possibly contains open assumption classes. Then the following two statements holds:*

(i) $\mathcal{D}$ does not fulfill the condition **AA** *if and only if $\mathcal{D}$ can be represented by a symbolic prooftree of the form*

$$\left.\begin{array}{c} \mathcal{D}_2 \\ (\tau_1' \to \tau_2' = \sigma_1' \to \sigma_2') \\ \mathcal{DC}_1 \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \;\; \text{ARROW } \textit{or} \text{ ARROW/FIX} \\ (\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2) \\ \mathcal{DC}_0 \end{array}\right\} \tag{C.5}$$

with $\tau_1, \tau_2, \tau_1', \tau_2', \sigma_1, \sigma_2, \sigma_1', \sigma_2' \in \mu Tp$ *such that* $\tau_1 \to \tau_2 \equiv_{ren} \tau_1' \to \tau_2'$ *and also* $\sigma_1 \to \sigma_2 \equiv_{ren} \sigma_1' \to \sigma_2'$ *holds, with derivation contexts* $\mathcal{DC}_0$ *and* $\mathcal{DC}_1$*, and with a subderivation* $\mathcal{D}_2$ *that contains at least one application of a rule different from* REN.

(ii) *If* $\mathcal{D}$ *does not fulfill the condition* **ADA** *if and only if* $\mathcal{D}$ *can be represented by a symbolic prooftree of the form* (C.5) *with* $\tau_1, \tau_2, \tau_1', \tau_2', \sigma_1, \sigma_2, \sigma_1', \sigma_2' \in \mu Tp$ *such that* $\tau_1 \to \tau_2 \equiv_{ren} \tau_1' \to \tau_2'$ *and* $\sigma_1 \to \sigma_2 \equiv_{ren} \sigma_1' \to \sigma_2'$ *holds, with derivation contexts* $\mathcal{DC}_0$ *and* $\mathcal{DC}_1$*, and with a subderivation* $\mathcal{D}_2$ *such that*

- $\mathcal{D}_2$ *contains at least one rule application different from* REN*, or*
- $\mathcal{D}_2$ *consists of a (possibly empty) sequence of* REN*-applications below a marked assumption that in* $\mathcal{D}$ *corresponds to an open marked assumption.*

Proof. The proposition is an obvious consequence of Definition C.3 and the fact mentioned previous to its statement that the assertion (C.3) in Definition C.3 can be replaced by the assertion (C.4) without changing this definition. □

It is easy to conclude from Proposition C.4, (ii), that a $\mathbf{HB}_{\mathbf{0}}^{=}$-derivation $\mathcal{D}$ contains a detour in the sense of Remark C.1 if and only if $\mathcal{D}$ does not fulfill the condition **ADA**. Now we define a "violation of the condition **ADA**" in a $\mathbf{HB}_{\mathbf{0}}^{=}$-derivation $\mathcal{D}$ to be such a formula occurrences in $\mathcal{D}$ that is the conclusion of a detour subderivation in the sense of Remark C.1. And similarly, we define "violations of the condition **AA**".

Definition C.5 (Violations of the conditions AA and ADA). Let $\mathcal{D}$ be a derivation in $\mathbf{HB}_{\mathbf{0}}^{=}$.

(i) Let $\mathcal{D}$ be a derivation in $\mathbf{HB}_{\mathbf{0}}^{=}$ that can be represented as a symbolic prooftree of the form (C.5) with $\tau_1, \tau_2, \tau_1', \tau_2', \sigma_1, \sigma_2, \sigma_1', \sigma_2' \in \mu Tp$ such that $\tau_1 \to \tau_2 \equiv_{\text{ren}}$ $\equiv_{\text{ren}} \tau_1' \to \tau_2'$ and $\sigma_1 \to \sigma_2 \equiv_{\text{ren}} \sigma_1' \to \sigma_2'$ holds, with derivation contexts $\mathcal{DC}_0$ and $\mathcal{DC}_1$, and with a subderivation $\mathcal{D}_2$ that contains at least one application of a rule different from REN (due to Proposition C.4 this means that $\mathcal{D}$ does not fulfill the condition **AA**). Then we say that the occurrence of the formula $\tau_1' \to \tau_2' = \sigma_1' \to \sigma_2'$ in (C.5) is *a violation of the condition* **AA** *in* $\mathcal{D}$.

(ii) Let $\mathcal{D}$ be a derivation in $\mathbf{HB}_{\mathbf{0}}^{=}$ that can be represented by a symbolic prooftree of the form (C.5) with $\tau_1, \tau_2, \tau_1', \tau_2', \sigma_1, \sigma_2, \sigma_1', \sigma_2' \in \mu Tp$ such that $\tau_1 \to \tau_2 \equiv_{\text{ren}}$ $\equiv_{\text{ren}} \tau_1' \to \tau_2'$ and $\sigma_1 \to \sigma_2 \equiv_{\text{ren}} \sigma_1' \to \sigma_2'$ holds, with derivation contexts $\mathcal{DC}_0$ and $\mathcal{DC}_1$, and with a subderivation $\mathcal{D}_2$ that contains at least one rule application different from REN or that consists of a (possibly empty) sequence of REN-applications below a marked assumption that in $\mathcal{D}$ corresponds to an open marked assumption (due to Proposition C.4 this means that $\mathcal{D}$ does not fulfill the condition **ADA**). Then we say that the occurrence of the formula $\tau_1' \to \tau_2' = \sigma_1' \to \sigma_2'$ in (C.5) is *a violation of the condition* **ADA** *in* $\mathcal{D}$.

⊠

The following proposition formalizes the following easy observation: In the system $\mathbf{HB}_{\mathbf{0}}^{=}$ every derivation $\mathcal{D}$ that does not violate the binding condition **AA** or equivalently, that satisfies **AA**, is actually very near to a derivation $\mathcal{D}'$ that satisfies **ADA** and that can be easily reached from $\mathcal{D}$. Transforming $\mathcal{D}$ to such a derivation $\mathcal{D}'$ involves hereby only the operations of extending $\mathcal{D}$ above some of its leaves by additional applications of REN and of binding back such additionally arising assumptions to existing applications of ARROW/FIX or of ARROW in $\mathcal{D}$, which in the case of existing applications of ARROW however means to change these into applications of ARROW/FIX.

Proposition C.6. (Removing violations of ADA from $\mathbf{HB}_{\mathbf{0}}^{=}$-derivations that fulfill AA). *Every derivation $\mathcal{D}$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ with possibly open assumption classes that fulfills the condition* **AA** *can effectively be transformed into a derivation $\mathcal{D}'$ that mimics $\mathcal{D}$ and that satisfies* **ADA**.

Sketch of Proof. Since this proposition follows from Proposition C.7 below, we only sketch the proof here, which proceeds by induction on the number $\#V_{\mathbf{ADA}}(\mathcal{D})$ of violations of the condition **ADA** in a derivation $\mathcal{D}$ in $\mathbf{HB}_{\mathbf{0}}^{=}$.

Let $\mathcal{D}$ be a derivation in $\mathbf{HB}_{\mathbf{0}}^{=}$ that fulfills **AA**, but not **ADA**. Then $\mathcal{D}$ contains violations of **ADA** that are not violations of **ADA**. These violations can successively be eliminated. By displaying an arbitrary violation of **ADA** in $\mathcal{D}$ (that hence is not also a violation of **AA**), an occurrence of the formula $\tau_1' \to \tau_2' = \sigma_1' \to \sigma_2'$, the derivation $\mathcal{D}$ can be represented as a prooftree of the form (C.5) with the recursive types, derivation contexts and derivations as in Proposition C.4, (ii); in particular, $\mathcal{D}_2$ contains only applications of REN below a marked assumption of the form $\tau_1'' \to \tau_2'' = \sigma_1'' \to \sigma_2''$, for some $\tau_1'', \tau_2'', \sigma_1'', \sigma_2'' \in \mu Tp$ such such that $\tau_1'' \to \tau_2'' \equiv_{\text{ren}} \tau_1 \to \tau_2$ and $\sigma_1'' \to \sigma_2'' \equiv_{\text{ren}} \sigma_1 \to \sigma_2$.

By extending $\mathcal{D}_2$ within $\mathcal{D}$ by an additional REN-application with the marked assumption $(\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2)^u$ as assumption, by changing the bottommost rule application in $\mathcal{DC}_1$ into an application of ARROW/FIX at which assumptions marked by u are discharged, and by some necessary marker renaming in $\mathcal{DC}_1$, $\mathcal{D}$ can be transformed into a derivation $\mathcal{D}^{(1)}$ that still fulfills **AA**, but that has one violation of **ADA** less than $\mathcal{D}$. All other violations of **ADA** in $\mathcal{D}^{(1)}$ can subsequently be eliminated in a similar way. □

It is also possible to transform effectively every given $\mathbf{HB}_{\mathbf{0}}^{=}$-derivation $\mathcal{D}$ with possibly open assumption classes into a derivation $\mathcal{D}'$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ with the same conclusion and the same or fewer open assumption classes as $\mathcal{D}$, such that $\mathcal{D}'$ satisfies **ADA**. In this case, however, discharging open assumptions appropriately is not sufficient. Here also all detours of the form as described in Remark C.1 have to be "axed out" successively in the manner as explained in this remark.

Proposition C.7 (Removing violations of ADA from $\mathbf{HB}_{\mathbf{0}}^{=}$-derivations). *Every derivation $\mathcal{D}$ in $\mathbf{HB}_{\mathbf{0}}^{=}$, which possibly contains open assumption classes, can effectively be transformed into a derivation $\mathcal{D}'$ that mimics $\mathcal{D}$ and that fulfills the condition* **ADA**.

Proof. This can be shown by induction on the number $\#V_{\mathbf{ADA}}(\mathcal{D})$ of violations of the condition **ADA** in a $\mathbf{HB}_{\mathbf{0}}^{=}$-derivation $\mathcal{D}$.

In the base case of the induction, where $\#V_{\mathbf{ADA}}(\mathcal{D}) = 0$, nothing has to be shown.

For treatment of the induction step, let now $\mathcal{D}$ be an arbitrary derivation in $\mathbf{HB}_{\mathbf{0}}^{=}$ with $\#V_{\mathbf{ADA}}(\mathcal{D}) = n > 0$. Then $\mathcal{D}$ can be represented by a symbolic prooftree of the form (C.4) with $\tau_1, \tau_2, \tau_1', \tau_2', \sigma_1, \sigma_2, \sigma_1', \sigma_2' \in \mu Tp$ such that $\tau_1 \to \tau_2 \equiv_{\text{ren}} \tau_1' \to \tau_2'$ and $\sigma_1 \to \sigma_2 \equiv_{\text{ren}} \sigma_1' \to \sigma_2'$ holds, with derivation contexts $\mathcal{DC}_0$ and $\mathcal{DC}_1$, and with a subderivation $\mathcal{D}_2$ such that the displayed occurrence of $\tau_1' \to \tau_2' = \sigma_1' \to \sigma_2'$ is a violation of the condition **ADA** in $\mathcal{D}$. By applying the kind of simplification explained in Remark C.1, $\mathcal{D}$ can be transformed into a derivation $\mathcal{D}^{(1)}$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ of the form

$$\begin{array}{c}
\dfrac{(\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2)^u}{(\tau_1' \to \tau_2' = \sigma_1' \to \sigma_2')}\ \mathrm{VAR} \\
\mathcal{DC}_1' \\
\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\ \mathrm{ARROW/FIX},\, u \\
(\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2) \\
\mathcal{DC}_0
\end{array}$$

for some assumption marker u and for a derivation context $\mathcal{DC}_1'$ that results from $\mathcal{DC}_1$ by renaming the assumption markers of the assumptions (possibly) discharged at the bottommost rule application in $\mathcal{DC}_1$ to u, such that $\mathcal{D}^{(1)}$ mimics $\mathcal{D}$ (since it has the same conclusion as $\mathcal{D}$, but equally many or less open assumption classes) and such that $\#V_{\mathbf{ADA}}(\mathcal{D}^{(1)}) < \#V_{\mathbf{ADA}}(\mathcal{D})$ holds. By applying the induction hypothesis, $\mathcal{D}^{(1)}$ can be transformed into a derivation $\mathcal{D}'$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ that mimics $\mathcal{D}^{(1)}$, that hence also mimics $\mathcal{D}$, and that fulfills the condition **ADA**. □

In the following remark we describe what we mean by redundancies of the kind (Rdcy2) mentioned earlier.

Remark C.8 (Redundancies of kind (Rdcy2) in $\mathbf{HB}_{\mathbf{0}}^{=}$-derivations). It follows from Lemma 7.1.1, (ii), that for all $\tau_0 \in \mu Tp$ and $\alpha \in TVar$ the following holds:

$$\left.\begin{array}{rl}
\alpha \downarrow \tau_0 \iff & \mathrm{nl}\mu\mathrm{b}(\,\tau_0[\mu\alpha.\,\tau_0/\alpha]\,) \;<\; \mathrm{nl}\mu\mathrm{b}(\mu\alpha.\,\tau_0) \\
\iff & \mathrm{nl}\mu\mathrm{b}(\,\tau_0[\mu\alpha.\,\tau_0/\alpha]\,) \;=\; \mathrm{nl}\mu\mathrm{b}(\mu\alpha.\,\tau_0) - 1\ .
\end{array}\right\} \tag{C.6}$$

Due to this, applications of the rule FOLD_l of the particular kind

$$\frac{\tau_0[\mu\alpha.\,\tau_0/\alpha] = \sigma}{\mu\alpha.\,\tau_0 = \sigma}\ \mathrm{FOLD}_l \quad (\text{where } \alpha \not\downarrow \tau_0), \tag{C.7}$$

where $\tau_0, \sigma \in \mu Tp$ and $\alpha \in TVar$, are precisely those applications of FOLD_l in which the number $\mathrm{nl}\mu\mathrm{b}(\cdot)$ of leading μ-bindings either stays the same or actually decreases in the recursive type that gets folded, i.e. from $\mathrm{nl}\mu\mathrm{b}(\tau_0[\mu\alpha.\,\tau_0/\alpha])$ to $\mathrm{nl}\mu\mathrm{b}(\mu\alpha.\,\tau_0)$. An analogous assertion holds clearly also for applications of the rule FOLD_r. We will call applications of FOLD_l of the form (C.7) as well as similar applications of FOLD_r "nlμb-decreasing" (see Definition C.9 below).

Such applications of FOLD_l or FOLD_r are the cause of redundancies in $\mathbf{HB}_{\mathbf{0}}^{=}$-derivations because they enable detour-loops like in a derivation of the form

$$\begin{array}{c}\vdots\\ \dfrac{\dfrac{\mu\alpha\alpha_1.\,\alpha = \sigma}{\mu\alpha_1\alpha\alpha_1.\,\alpha = \sigma}\,\mathrm{FOLD}_l}{\mu\alpha\alpha_1.\,\alpha = \sigma}\,\mathrm{FOLD}_l \qquad (*)\end{array}$$

as well as in similar derivations. It is hereby just the application of FOLD_l labeled by (*) that is of the form (C.7), due to $\alpha \not\downarrow \mu\alpha_1.\,\alpha$; the previous application of FOLD_l is of a different kind because $\alpha_1 \downarrow \mu\alpha\alpha_1.\,\alpha$ holds as a consequence of $\alpha_1 \notin \mathrm{fv}(\mu\alpha\alpha_1.\,\alpha)$.

Applications of FOLD_l of the form (C.7), and analogous applications of FOLD_r will be excluded from appearing in such $\mathbf{HB}_{\mathbf{0}}^{=}$-derivations $\mathcal{D}$ for which of the main theorem in this appendix, Theorem C.11, states a bound on its depth $|\mathcal{D}|$. However, we will see first that as a consequence of Lemma C.10 below, nlμb-decreasing applications of $\mathrm{FOLD}_{l/r}$ can always be removed from derivations in $\mathbf{HB}_{\mathbf{0}}^{=}$ *without open assumptions* in a very easy way.

The explanations in Remark C.8 above justify the following definition.

Definition C.9 (nlμb-decreasing $\mathrm{FOLD}_{l/r}$-applications). Let $\tau_0, \sigma \in \mu Tp$, $\alpha \in \mathit{TVar}$, and let an application of FOLD_l or of FOLD_r of the respective form

$$\dfrac{\tau_0[\mu\alpha.\,\tau_0/\alpha] = \sigma}{\mu\alpha.\,\tau_0 = \sigma}\,\mathrm{FOLD}_l \qquad \text{or} \qquad \dfrac{\sigma = \tau_0[\mu\alpha.\,\tau_0/\alpha]}{\sigma = \mu\alpha.\,\tau_0}\,\mathrm{FOLD}_r \tag{C.8}$$

be given. We call this application *nlμb-decreasing* if and only if $\alpha \not\downarrow \tau_0$ holds. ⊠

The statement of the following lemma is slightly more general than the assertion mentioned at the end of Remark C.8. It states that redundancies of the kind (Rdcy2) can always be removed from derivations without assumptions in the extension of $\mathbf{HB}_{\mathbf{0}}^{=}$ with the rules SYMM, TRANS, and $\{\mathrm{TRANS}_k\}_k$ (the generalized transitivity rules introduced in Definition 8.2.8, Section 8.2) with the result of respective mimicking derivations. For use in this lemma, we define the set

$$\begin{aligned}\mathrm{BOT} =_{\mathrm{def}} & \left\{\mu\alpha_1\ldots\alpha_n.\,\bot \mid n \in \omega,\ \alpha_1,\ldots,\alpha_n \in \mathit{TVar}\right\} \ \cup \\ & \cup\ \left\{\mu\alpha_1,\ldots,\alpha_n.\,\alpha_i \mid n \in \omega,\ \alpha_1,\ldots,\alpha_n \in \mathit{TVar},\ i \in \{1,\ldots,n\}\right\}.\end{aligned}$$

It is easy to prove that BOT is the set of all recursive types that have $\bot$ as their leading symbol.

Lemma C.10. *The following three assertions hold:*

(i) For all $\rho, \chi \in \mathrm{BOT}$, a derivation $\mathcal{D}_{(\rho=\chi)}$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ with conclusion $\rho = \chi$ and without open assumptions can effectively be constructed such that $\mathcal{D}_{(\rho=\chi)}$ contains neither nlμb-decreasing applications of rules $\mathrm{FOLD}_{l/r}$ *nor applications of* REN.

(ii) *Let $\mathcal{D}$ be a derivation in* $\mathbf{HB}_{\mathbf{0}}^{=} + \mathrm{SYMM} + \mathrm{TRANS} + \{\mathrm{TRANS}_k\}_k$ *of the form*

$$\begin{array}{c} \mathcal{D}_a \\ (\rho = \chi) \\ \mathcal{DC}_0 \\ \tau = \sigma \end{array} \tag{C.9}$$

with conclusion $\tau = \sigma$*, for some* $\tau, \sigma \in \mu Tp$*, without open assumptions and where* $\rho = \chi$ *is an occurrence of an equation between recursive types such that* $\rho \in \mathrm{BOT}$ *or* $\chi \in \mathrm{BOT}$ *is the case.*

Then it follows that the subderivation $\mathcal{D}_a$ *of* $\mathcal{D}$ *does not contain open assumptions and that* both $\rho \in \mathrm{BOT}$ *and* $\chi \in \mathrm{BOT}$ *is the case.*

(iii) *Every derivation* $\mathcal{D}$ *in* $\mathbf{HB}_{\mathbf{0}}^{=} + \mathrm{SYMM} + \mathrm{TRANS} + \{\mathrm{TRANS}_k\}_k$ *without open assumptions can effectively be transformed into a derivation* $\mathcal{D}'$ *in* $\mathbf{HB}_{\mathbf{0}}^{=} + \mathrm{SYMM} + \mathrm{TRANS} + \{\mathrm{TRANS}_k\}_k$ *that mimics* $\mathcal{D}$ *and that does not contain nlμb-decreasing applications of* $\mathrm{FOLD}_{l/r}$.

Proof. We demonstrate the assertions (i), (ii), and (iii) of the lemma in the three items (a), (b), and (c) below, respectively.

(a) Instead of proving the assertion in item (i) of the lemma formally, we only give a typical example for a derivation $\mathcal{D}_{(\rho=\chi)}$ in $\mathbf{HB}_{\mathbf{0}}^{=}$; it is easy to convert the idea exhibited in this example into a formal proof. For the two recursive types $\mu\alpha_2\alpha_1\alpha_2.\,\alpha_1$ and $\mu\beta\gamma.\,\bot$ in BOT the following is a desired derivation $\mathcal{D}_{(\mu\alpha_2\alpha_1\alpha_2.\,\alpha_1=\mu\beta\gamma.\,\bot)}$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ without open assumptions and without nlμb-decreasing applications of $\mathrm{FOLD}_{l/r}$:

$$\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{(\mathrm{REFL})}{\bot = \bot}}{\mu\alpha_1\alpha_2.\,\alpha_1 = \bot}\,(\mu{-}\bot)_l^{\bot\mathrm{der}}}{\mu\alpha_2\alpha_1\alpha_2.\,\alpha_1 = \bot}\,\mathrm{FOLD}_l}{\mu\alpha_2\alpha_1\alpha_2.\,\alpha_1 = \mu\gamma.\,\bot}\,\mathrm{FOLD}_r}{\mu\alpha_2\alpha_1\alpha_2.\,\alpha_1 = \mu\beta\gamma.\,\bot}\,\mathrm{FOLD}_r$$

(b) Assertion (ii) of the lemma can be shown in a straightforward way by induction on the depth $|\mathcal{D}_a|$ of the subderivation $\mathcal{D}_a$ in a derivation $\mathcal{D}$ in $\mathbf{HB}_{\mathbf{0}}^{=} + \mathrm{SYMM} + \mathrm{TRANS} + \{\mathrm{TRANS}_k\}_k$ without open assumption classes that is of the form (C.9) such that $\rho \in \mathrm{BOT}$ or $\chi \in \mathrm{BOT}$. In the induction step the easy verifiable fact is used that the set BOT is closed under conversions with respect to the reduction relation $\rightarrow_{\mathrm{r/o\text{-}u}(\mu\bot)'}$ from Definition 5.3.5.

(c) Assertion (iii) of the lemma can be shown by induction on the number of nlμb-decreasing applications of $\mathrm{FOLD}_{l/r}$ in a derivation $\mathcal{D}$ in $\mathbf{HB}_{\mathbf{0}}^{=} + \mathrm{SYMM} + \mathrm{TRANS} + \{\mathrm{TRANS}_k\}_k$.

The base case of the induction is obvious.

For the induction step, let $\mathcal{D}$ be a derivation in the system $\mathbf{HB}_{\mathbf{0}}^{=} + \text{SYMM} + {} + \text{TRANS} + \{\text{TRANS}_k\}_k$ without open assumptions that contains at least one nlμb-decreasing application of $\text{FOLD}_{l/r}$. Then, with respect to a particular nlμb-decreasing application of FOLD_l that is exhibited, $\mathcal{D}$ can be represented as of the form

$$\left.\begin{array}{c} \mathcal{D}_a \\ \rho_0[\mu\alpha.\,\rho_0/\alpha] = \chi \\ \cdots\cdots\cdots\cdots\cdots\cdots \ \text{FOLD}_l \\ (\mu\alpha.\,\rho_0 = \chi) \\ \mathcal{DC}_0 \\ \tau = \sigma \end{array}\right\} \tag{C.10}$$

for some $\tau, \sigma, \mu\alpha.\,\rho_0, \chi \in \mu Tp$ such that $\alpha \downarrow \rho_0$ holds, and for a subderivation $\mathcal{D}_a$ and a derivation context $\mathcal{DC}_0$; in case that $\mathcal{D}$ contains only nlμb-decreasing applications of FOLD_r, it can be argued analogously as below. Since $\alpha \not\downarrow \rho_0$ holds, it follows by Lemma 7.1.1, (i), that $\rho_0 \equiv \mu\alpha_1 \ldots \alpha_n.\,\alpha$ for some $\alpha_1, \ldots, \alpha_n \in TVar$ and $\alpha \not\equiv \alpha_1, \ldots, \alpha_n$. This entails $\mu\alpha.\,\rho_0 \in \text{BOT}$. Assertion (ii) of the lemma now implies that also $\chi \in \text{BOT}$ holds. Hence by assertion (i) of the lemma, a derivation $\mathcal{D}_{(\mu\alpha.\,\rho_0=\chi)}$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ exists that mimics $\mathcal{D}_a$. Therefore $\mathcal{D}_a$ can be replaced within $\mathcal{D}$ by $\mathcal{D}_{(\mu\alpha.\,\rho_0=\chi)}$, with the result of a derivation $\tilde{\mathcal{D}}'$ of the form

$$\begin{array}{c} \mathcal{D}_{(\mu\alpha.\,\rho_0=\chi)} \\ (\mu\alpha.\,\rho_0 = \chi) \\ \mathcal{DC}_0 \\ \tau = \sigma \end{array}$$

that mimics $\mathcal{D}$ and that contains at least one nlμb-decreasing application of $\text{FOLD}_{l/r}$ less than $\mathcal{D}$. Now it follows by the induction hypothesis that $\tilde{\mathcal{D}}'$ can effectively be transformed into a derivation $\mathcal{D}'$ in $\mathbf{HB}_{\mathbf{0}}^{=} + \text{SYMM} + \text{TRANS} + {} + \{\text{TRANS}_k\}_k$ that mimics $\tilde{\mathcal{D}}'$, and hence that also mimics $\mathcal{D}$, such that $\mathcal{D}'$ does not contain any nlμb-decreasing applications of $\text{FOLD}_{l/r}$.

□

Now we formulate, and then we prove, the main theorem of this appendix. It states that, for a derivation $\mathcal{D}$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ without redundancies of the kinds (Rdcy1), (Rdcy2), and (Rdcy3), the depth $|\mathcal{D}|$ of $\mathcal{D}$ is essentially bounded by a multiple of the product of the sizes of the recursive types τ and σ in the conclusion $\tau = \sigma$ of $\mathcal{D}$.

Theorem C.11 (An upper bound for 'not redundant' $\mathbf{HB}_{\mathbf{0}}^{=}$-derivations). *Let $\mathcal{D}$ be a derivation in $\mathbf{HB}_{\mathbf{0}}^{=}$ with possibly open assumption classes such that the following three conditions hold:*

(i) $\mathcal{D}$ does not contain two successive applications of the rule REN.

(ii) $\mathcal{D}$ *does not contain nlμb-decreasing applications of* $\mathrm{FOLD}_{l/r}$.

(iii) $\mathcal{D}$ *fulfills the condition* **AA**.

Then the depth of $\mathcal{D}$ *can be bounded by*

$$|\mathcal{D}| < 2\left((s(\tau)+1)(s(\sigma)+1)+2|\tau|+2|\sigma|+2\right) . \tag{C.11}$$

This theorem will be *proved* below on p. 406 after stating and proving the following lemma, which asserts that at least every $\mathbf{HB}_{\mathbf{0}}^{=}$-derivation $\mathcal{D}$ *without open assumptions* can be transformed into a mimicking derivation $\mathcal{D}'$ for $\mathcal{D}$ that fulfills the hypotheses of Theorem C.11.

Lemma C.12. *Every derivation* $\mathcal{D}$ *in* $\mathbf{HB}_{\mathbf{0}}^{=}$ *without open assumptions can effectively be transformed into a derivation* $\mathcal{D}'$ *in* $\mathbf{HB}_{\mathbf{0}}^{=}$ *without open assumptions and with the same conclusion as* $\mathcal{D}$ *such that* $\mathcal{D}'$ *fulfills the conditions (i), (ii) and (iii) in Theorem C.11, and such that* $\mathcal{D}'$ *furthermore satisfies the condition* **ADA**.

Proof. Let $\mathcal{D}$ be an arbitrary derivation in $\mathbf{HB}_{\mathbf{0}}^{=}$ without open assumptions.

In a first step, $\mathcal{D}$ can effectively be transformed, due to Lemma C.10, (iii), and more precisely, due to the transformation described in the proof of this statement, into a derivation $\mathcal{D}^{(1)}$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ that fulfills condition (ii) in Theorem C.11 and that mimics $\mathcal{D}$.

In a second step, $\mathcal{D}^{(1)}$ can effectively be transformed, due to Proposition C.7, into a derivation $\mathcal{D}^{(2)}$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ that mimics $\mathcal{D}^{(1)}$ as well as $\mathcal{D}$, and that fulfills the condition **ADA**. Since in the transformation described in the proof of Proposition C.7 no applications of rules $\mathrm{FOLD}_{l/r}$ are introduced, $\mathcal{D}^{(2)}$ fulfills, as does $\mathcal{D}^{(1)}$, the condition (ii) in Theorem C.11.

By 'contracting', in a third step, each sequence of two or more successive applications of REN into a single application of REN, a derivation $\mathcal{D}'$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ that mimics $\mathcal{D}^{(2)}$, that hence also mimics $\mathcal{D}$, and that fulfills the conditions (i), (ii), and (iii) in Theorem C.11 can effectively be found (none of the properties (ii) and (iii) in Theorem C.11 are affected by 'contracting' REN-applications); furthermore $\mathcal{D}'$ obviously still fulfills the condition **ADA**.

In this way we have effectively produced from $\mathcal{D}$ a derivation $\mathcal{D}'$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ without open assumptions and with the same conclusion as $\mathcal{D}$ such that $\mathcal{D}'$ fulfills the hypotheses of Theorem C.11 and such that $\mathcal{D}'$ even satisfies the condition **ADA**.

□

Proof of Theorem C.11. We will prove the theorem by showing the following logically equivalent statement: For every derivation $\mathcal{D}$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ with possibly open assumption classes that satisfies the conditions (i) and (ii) in the theorem and that has a depth greater or equal to the expression on the right-hand side of (C.11), it is the case that the condition **AA** is violated, i.e. the condition (iii) in the theorem does not hold for $\mathcal{D}$.

For showing this, we let $\tau, \sigma \in \mu Tp$ be arbitrary and we let $\mathcal{D}$ be an arbitrary derivation in $\mathbf{HB}_{\mathbf{0}}^{=}$ with conclusion $\tau = \sigma$, with possibly open assumption classes, and with the properties that $\mathcal{D}$ does not contain two successive applications of REN nor any applications of $\mathrm{FOLD}_{l/r}$ of the form excluded by condition (ii) in the theorem, and that

$$n =_{\mathrm{def}} |\mathcal{D}| \geq\ 2\left((\mathrm{s}(\tau)+1)(\mathrm{s}(\sigma)+1)+2|\tau|+2|\sigma|+2\right)\ . \tag{C.12}$$

holds. We will show that $\mathcal{D}$ violates the condition **AA**.

Assumption (C.12) means that there exists a thread from one of the leaves of $\mathcal{D}$ downwards to its conclusion of $\mathcal{D}$ that is of length n. We choose such a thread Θ and write it as

$$\left.\begin{array}{c} \dfrac{\tau_n = \sigma_n}{\tau_{n-1} = \sigma_{n-1}}\ R_n \\ \\ \vdots \\ \\ \dfrac{\tau_1 = \sigma_1}{\tau_0 = \sigma_0}\ R_0 \end{array}\right\}\ \Theta \tag{C.13}$$

with some $\tau_0, \ldots, \tau_n, \sigma_0, \ldots, \sigma_n \in \mu Tp$ such that $\tau_0 \equiv \tau$ and $\sigma_0 \equiv \sigma$ and where the dotted lines in Θ indicate the respective rule applications that are passed on Θ in $\mathcal{D}$. In (C.13) we have also labeled these dotted lines with the names of the corresponding rules R_i, where $i \in \{0, \ldots, n\}$, respective applications of which are passed on Θ in $\mathcal{D}$.

We define the function $f : \{0, 1, \ldots, n\} \to \omega$ inductively by the clauses

$$\left.\begin{array}{l} f(0) =_{\mathrm{def}} 0\ , \\[1ex] \text{and for all } i \in \{0, \ldots, n-1\}\text{:} \\[1ex] \quad f(i+1) =_{\mathrm{def}} \begin{cases} f(i) & \ldots \text{ if } R_i = \mathrm{REN} \\ f(i)+1 & \ldots \text{ else}\ . \end{cases} \end{array}\right\} \tag{C.14}$$

Now we set

$$\tilde{n} =_{\mathrm{def}} f(n)\ . \tag{C.15}$$

By this definition f is monotonously increasing and has range $\mathrm{ran}(f) = \{0, 1, \ldots, \tilde{n}\}$. Its definition can be understood as follows: The value $f(i)$ of f, for an arbitrary argument $i \in \{0, 1, \ldots, n\}$, is precisely the number of rule applications different from REN that are passed on Θ in $\mathcal{D}$ below the occurrence of the equation $\tau_i = \sigma_i$ in Θ. Hence $\tilde{n}$ is the total number of all applications of non-REN-rules that are passed on the thread Θ in $\mathcal{D}$. Since $\mathcal{D}$ does not contain any two successive applications of REN by assumption, it follows that:

$$(\,\forall i \in \{0, \ldots, n\}\,)\ \left[\, i\ \leq\ 2f(i)+1 \,\right]\ . \tag{C.16}$$

We also define the following function

$$\begin{aligned} g : \{0, 1, \ldots, \tilde{n}\}\ &\longrightarrow\ \{0, 1, \ldots, n\} \\ i^*\ &\longmapsto\ \max(f^{-1}(\{i^*\}))\ , \end{aligned} \tag{C.17}$$

which is well-defined since $\mathrm{ran}(f) = \{0, 1, \ldots, \tilde{n}\}$. As an aside, we note that, for all $i^* \in \{0, \ldots, \tilde{n}\}$, $|f^{-1}(\{i^*\})| \leq 2$ holds, since by assumption (condition (i) of the theorem holds for $\mathcal{D}$) there are no two successive applications of REN in $\mathcal{D}$. It follows from the definition of g that this function 'inverts' f in the sense that $f \circ g = id_{\{0,\ldots,\tilde{n}\}}$ holds. And furthermore, g is strictly increasing as a consequence of its definition together with the fact that f is non-decreasing.

We now let

$$\tilde{n}_0 =_{\mathrm{def}} (\mathrm{s}(\tau) + 1)(\mathrm{s}(\sigma) + 1) \ . \tag{C.18}$$

Since by the definition of n in (C.12) $2\,\tilde{n}_0 + 1 < n$ follows, we find from (C.16) that

$$2\,\tilde{n}_0 + 1 \ \leq 2\,f(2\,\tilde{n}_0 + 1) + 1 \quad \text{and from this}^3 \quad \tilde{n}_0 \ \leq \ f(2\,\tilde{n}_0 + 1) \ \leq \ f(n) \ = \ \tilde{n} \ .$$

Because this entails $\tilde{n}_0 \in \mathrm{dom}(g)$, we are able to define

$$n_0 =_{\mathrm{def}} g(\tilde{n}_0) \ . \tag{C.19}$$

Another application of (C.16), and the fact that g inverts f leads us to:

$$n_0 \ = \ g(\tilde{n}_0) \ \leq \ 2\,f(g(\tilde{n}_0)) + 1 \ = \ 2\,\tilde{n}_0 + 1 \ . \tag{C.20}$$

By Proposition 5.1.17, the fact that the proof system $\mathbf{HB_0^=}$ obeys the subformula property SP_1, it follows that the derivation $\mathcal{D}$ fulfills SP_1 (cf. Definition 5.1.16). This implies

$$(\forall i \in \{0, 1, \ldots, n\}) \left[\, \tau_i \sqsubseteq' \tau_0 \ \ \& \ \ \sigma_i \sqsubseteq' \sigma_0 \,\right] ,$$

i.e. that, for every equation $\tau_i = \sigma_i$ in the thread Θ, τ_i is a $\rightarrow_{\mathrm{roud}\perp}$-generated subterm of τ_0 and σ_i is a $\rightarrow_{\mathrm{roud}\perp}$-generated subterm of σ_0. Due to this, we find the following, using Corollary 3.9.24:

$$\left.\begin{aligned} &\left|\{\ ([\tau_{g(i^*)}]_{\equiv_{\mathrm{ren}}}\,,\,[\sigma_{g(i^*)}]_{\equiv_{\mathrm{ren}}}) \ \mid \ i^* \in \{0, 1, \ldots, \tilde{n}_0\}\ \}\right| \ \leq \\ &\qquad \leq \ \left|\{\ ([\tau_i]_{\equiv_{\mathrm{ren}}}\,,\,[\sigma_i]_{\equiv_{\mathrm{ren}}}) \ \mid \ i \in \{0, 1, \ldots, n_0\}\ \}\right| \ \leq \\ &\qquad \leq \ |\mathcal{G}'_*(\tau) \times \mathcal{G}'_*(\sigma)| \ \leq \ (\mathrm{s}(\tau) + 1)\,(\mathrm{s}(\sigma) + 1) \ = \ \tilde{n}_0 \ . \end{aligned}\ \right\} \tag{C.21}$$

Now due to the pidgin-hole principle it follows from (C.21) that

$$(\,\exists\, 0 \leq i^* < j^* \leq \tilde{n}_0\,) \left[\, \tau_{g(i^*)} \equiv_{\mathrm{ren}} \tau_{g(j^*)} \ \ \& \ \ \sigma_{g(i^*)} \equiv_{\mathrm{ren}} \sigma_{g(j^*)} \,\right] . \tag{C.22}$$

Via a translation $i^* \mapsto g(i^*)$, this implies the assertion

$$(\,\exists\, 0 \leq i < j \leq n_0\,) \left[\, f(i) \neq f(j) \ \ \& \ \ \tau_i \equiv_{\mathrm{ren}} \tau_j \ \ \& \ \ \sigma_i \equiv_{\mathrm{ren}} \sigma_j \,\right] \tag{C.23}$$

because g is strictly increasing. Due to this, we are entitled to choose i and j as follows: Let $(i, j) \in \{0, 1, \ldots, n_0\} \times \{0, 1, \ldots, n_0\}$ be the minimal possible pair with the property

$$i < j \ \text{ and } \ f(i) < f(j) \ \text{ and} \tag{C.24}$$

$$\tau_i \equiv_{\mathrm{ren}} \tau_j \ \text{ and } \ \sigma_i \equiv_{\mathrm{ren}} \sigma_j \ . \tag{C.25}$$

[3] By taking a closer look it is possible to conclude that also $\tilde{n}_0 < \tilde{n}$ holds. This stronger statement is not necessary at this stage, but it will be shown and used below (cf. (C.30)).

Since $f(i) < f(j)$, there is at least one application of a rule $\neq$ REN passed on Θ in $\mathcal{D}$ between the occurrences of $\tau_j = \sigma_j$ and of $\tau_i = \sigma_i$; the thread Θ can thus be written as of the form

$$\left.\begin{array}{l} \quad\tau_n = \sigma_n \\ \cdots\cdots\cdots\cdots\; R_n \\ \tau_{n-1} = \sigma_{n-1} \\ \qquad\vdots \\ \quad\tau_j = \sigma_j \\ \;\vdots\vdots\vdots\vdots\vdots\vdots\vdots\vdots\;\text{at least one rule} \neq \text{REN is passed} \\ \quad\tau_i = \sigma_i \\ \qquad\vdots \\ \quad\tau_1 = \sigma_1 \\ \;\cdots\cdots\cdots\; R_0 \\ \quad\tau = \sigma \end{array}\right\}\Theta \tag{C.26}$$

where we know (C.25) about τ_i, τ_j and σ_i, σ_j.

It is not possible that on the thread Θ an application of $(\mu\!-\!\perp)^{\perp\mathrm{der}}_{l/r}$ would be passed in $\mathcal{D}$ between the equalities $\tau_j = \sigma_j$ and $\tau_i = \sigma_i$: if namely, for example, an application of $(\mu\!-\!\perp)^{\perp\mathrm{der}}_{l}$ were passed in $\mathcal{D}$ on this section of Θ, then by the fulfilledness of the subformula property SP_1 for $\mathcal{D}$ it would follow $\perp \twoheadrightarrow_{\mathrm{roud}\perp} \tau_j$ and $\tau_i \twoheadrightarrow_{\mathrm{roud}\perp} \mu\alpha\alpha_1 \ldots \alpha_n.\,\alpha$, for some $\alpha, \alpha_1, \ldots, \alpha_n \in TVar$. This implies $\tau_j \equiv \perp$ and $\tau_i \not\equiv \perp$, and hence $\tau_i \not\equiv_{\mathrm{ren}} \tau_j$ in contradiction with (C.25).

Furthermore it is also not possible that *all* applications of rules different from REN that are passed in this part of the thread Θ are applications of $\mathrm{FOLD}_{l/r}$. This is a consequence of the fact that $\mathcal{D}$ does not contain such applications of $\mathrm{FOLD}_{l/r}$ in which the number $\mathrm{nl}\mu\mathrm{b}(\cdot)$ of leading μ-bindings stayed the same or decreased in the recursive type that gets folded (this holds, in view of cf. Remark C.8 and in particular in view of (C.6), because condition (ii) of Theorem C.11 is true for $\mathcal{D}$) and that therefore each application of $\mathrm{FOLD}_{l/r}$ in $\mathcal{D}$ strictly increases by 1 the number of leading μ-bindings in the recursive type that gets folded. Suppose namely that all applications of rules different from REN in the part of Θ in $\mathcal{D}$ between $\tau_j = \sigma_j$ and $\tau_i = \sigma_i$ are applications of rules $\mathrm{FOLD}_{l/r}$. Since by what we already know about Θ, there is at least one application of a rule different from REN passed in $\mathcal{D}$ on this section of the thread Θ, it would follow that $\mathrm{nl}\mu\mathrm{b}(\tau_i) > \mathrm{nl}\mu\mathrm{b}(\tau_j)$ or $\mathrm{nl}\mu\mathrm{b}(\sigma_i) > \mathrm{nl}\mu\mathrm{b}(\sigma_j)$ true in contradiction with (C.25) in view of Proposition 3.5.6.

From this we conclude that there must exist at least one application of a $\mathbf{HB}_{\mathbf{0}}^{=}$-rule different from $(\mu\!-\!\perp)^{\perp\mathrm{der}}_{l/r}$, $\mathrm{FOLD}_{l/r}$ and REN in the part between the occurrences of the formulas $\tau_i = \sigma_i$ and $\tau_j = \sigma_j$ in the thread Θ. This part must therefore contain a rule ARROW or ARROW/FIX and can hence be represented as of the form

$$\left.\begin{array}{cl}
\tau_j = \sigma_j & \\
\vdots & \\
\tau_{k+1} = \sigma_{k+1} & \\
\cdots\cdots\cdots\cdots & \text{ARROW or ARROW/FIX} \\
\tau_k = \sigma_k & \\
::::::::: & (\text{REN}^*;\ \text{FOLD}_{l/r};\ \text{REN}^*)^m \\
\tau_i = \sigma_i &
\end{array}\right\}\ \Theta \tag{C.27}$$

where $k \in \{i,\, i+1, \ldots, j\}$

$$m = \text{nl}\mu\text{b}(\tau_i) + \text{nl}\mu\text{b}(\sigma_i) \tag{C.28}$$

(that m is precisely the number of applications of $\text{FOLD}_{l/r}$ in the sequence of rules $R_i, \ldots, R_{l+1}$ follows from the fact that the number $\text{nl}\mu\text{b}(\cdot)$ is exactly decreased by 1 in the recursive type that gets folded in each application of $\text{FOLD}_{l/r}$ in $\mathcal{D}$ and from $\text{nl}\mu\text{b}(\tau_k) + \text{nl}\mu\text{b}(\sigma_k) = 0$). Since $\tau_i, \sigma_i \not\equiv \bot$ holds (both τ_i and σ_i must now at least contain one occurrence of the symbol → because this is true for the types τ_k and σ_k in the conclusion of an application of ARROW or ARROW/FIX in $\mathcal{D}$), it follows by the fulfilledness of the subformula property SP_1 for $\mathcal{D}$ (Proposition 5.1.17) that $\tau_i \sqsubseteq' \tau_0 \equiv \tau$ and $\sigma_i \sqsubseteq' \sigma_0 \equiv \sigma$. By Lemma 3.9.25 (from Chapter 3, Section 3.1) we find that

$$m \;=\; \text{nl}\mu\text{b}(\tau_i) + \text{nl}\mu\text{b}(\sigma_i) \;\leq\; 2(|\tau| + |\sigma|)\ . \tag{C.29}$$

From (C.12), (C.20) and (C.29) we can conclude that

$$\left.\begin{array}{rcl}
n - n_0 & \geq & 2\,\big((\text{s}(\tau)+1)\,(\text{s}(\sigma)+1) + 2|\tau| + 2|\sigma| + 2\,\big) - \\
& & -\,\big(\,2\,(\text{s}(\tau)+1)\,(\text{s}(\sigma)+1) + 1\,\big) \\
& = & 2\,(2|\tau| + 2|\sigma| + 2) - 1 \\
& \geq & 2\,(m+1) + 1\ .
\end{array}\right\} \tag{C.30}$$

But this implies now that there are at least $m+1$ further applications different from REN passed in $\mathcal{D}$ on Θ before (i.e. above) the occurrence of $\tau_j = \sigma_j$. Due to (C.25) and Proposition 3.5.6 we have also $\text{nl}\mu\text{b}(\tau_j) + \text{nl}\mu\text{b}(\sigma_j) = m$. It follows that the first m applications of non-REN-rules passed in $\mathcal{D}$ on the thread Θ when going in upwards direction are applications of $\text{FOLD}_{l/r}$, in which the sum $\text{nl}\mu\text{b}(\tilde{\tau}) + \text{nl}\mu\text{b}(\tilde{\sigma})$ for the occurring formulas $\tilde{\tau} = \tilde{\sigma}$ respectively decreases by precisely one (since $\mathcal{D}$ fulfills assumption (ii) of the theorem, $\mathcal{D}$ does not contain nlμb-decreasing applications of

$\mathrm{FOLD}_{l/r}$). Thus the thread Θ must actually be of the form

$$\left.\begin{array}{cl}
\tau_n = \sigma_n & \\
\cdots\cdots\cdots\cdots\cdots & R_n \\
\tau_{n-1} = \sigma_{n-1} & \\
\vdots & \\
\tau_{l+1} = \sigma_{l+1} & \\
\cdots\cdots\cdots\cdots\cdots & \text{ARROW or ARROW/FIX} \\
\boldsymbol{\tau_l = \sigma_l} & \\
:::::::::: & \text{REN}^*,\ (\text{FOLD}_{l/r})^m \\
\tau_j = \sigma_j & \\
\vdots & \\
\tau_{k+1} = \sigma_{k+1} & \\
\cdots\cdots\cdots\cdots\cdots & \text{ARROW or ARROW/FIX} \\
\boldsymbol{\tau_k = \sigma_k} & \\
::::::::::: & (\text{REN}^*;\ \text{FOLD}_{l/r};\ \text{REN}^*)^m \\
\tau_i = \sigma_i & \\
\vdots & \\
\tau_1 = \sigma_1 & \\
\cdots\cdots\cdots\cdots\cdots\cdots\cdots & R_0 \\
(\tau \equiv)\ \tau_0 = \sigma_0\ (\equiv \sigma) &
\end{array}\right\}\ \Theta \qquad \text{(C.31)}$$

from which we can also read that $\tau_j \twoheadrightarrow_{\text{ren/out-unf}} \tau_l$ and $\sigma_j \twoheadrightarrow_{\text{ren/out-unf}} \sigma_l$. From this and from (C.25) as well as from $\tau_i \twoheadrightarrow_{\text{ren/out-unf}} \tau_k$ and $\sigma_i \twoheadrightarrow_{\text{ren/out-unf}} \sigma_k$ we read that furthermore $\tau_l \twoheadleftarrow\!\!\twoheadrightarrow_{\text{ren/out-unf}} \tau_k$ and $\sigma_l \twoheadleftarrow\!\!\twoheadrightarrow_{\text{ren/out-unf}} \sigma_k$. Since both of $\tau_k = \sigma_k$ and $\tau_l = \sigma_l$ are the conclusions of applications of ARROW or of ARROW/FIX in $\mathcal{D}$, we get that $\mathrm{nl}\mu\mathrm{b}(\tau_k) = \mathrm{nl}\mu\mathrm{b}(\tau_l) = \mathrm{nl}\mu\mathrm{b}(\sigma_k) = \mathrm{nl}\mu\mathrm{b}(\sigma_l) = 0$ holds. Now because of $\twoheadleftarrow\!\!\twoheadrightarrow_{\text{ren/out-unf}} \subseteq \twoheadleftarrow\!\!\twoheadrightarrow_{\text{r/o-u}(\mu\bot)'}$, Lemma 5.3.6, (iii), implies that

$$\tau_k \equiv_{\text{ren}} \tau_l \;\text{ and }\; \sigma_k \equiv_{\text{ren}} \sigma_l\ . \qquad \text{(C.32)}$$

But the fulfilledness of (C.32) with respect to the thread Θ, which we could depict as in (C.31), implies that $\mathcal{D}$ violates the condition **AA**. This is because the occurrence of $\tau_l = \sigma_l$ on Θ in $\mathcal{D}$ is a violation of **AA**: There is an occurrence of a formula $\tau_k = \sigma_k$ with respective variants τ_k and σ_k of τ_l and of σ_l deeper down in $\mathcal{D}$ that is separated from the occurrence of $\tau_l = \sigma_l$ by certainly one application of ARROW or ARROW/FIX; and the occurrence of $\tau_l = \sigma_l$ on Θ in $\mathcal{D}$ is *not* associated with an assumption of $\mathcal{D}$ because it is the conclusion of an application of ARROW or ARROW/FIX in $\mathcal{D}$.

Thus we have eventually shown that $\mathcal{D}$ violates the condition **AA**.

□

In the following theorem we formulate a statement that is slightly stronger than Theorem C.11 and that we have actually shown by our proof above for Theorem C.11.

Theorem C.13 (Strengthening of Theorem C.11). *Let $\mathcal{D}$ be a derivation in $\mathbf{HB}_0^{=}$ with conclusion $\tau = \sigma$ and with possibly open assumption classes. Furthermore let $\mathcal{D}$ be such that it fulfills the properties (i) and (ii) of Theorem C.11.*

Then for all threads Θ in the prooftree $\mathcal{D}$ from a formula in one of its leaves downwards to its conclusion that are of length

$$|\Theta| \geq \ 2\left((s(\tau)+1)(s(\sigma)+1)+2|\tau|+2|\sigma|+2\right)\,, \tag{C.33}$$

the following holds: There occurs a violation of the condition **ADA** *in $\mathcal{D}$ on Θ.*

Proof. We have shown this theorem implicitly in our proof of Theorem C.11. □

In view of Lemma C.12, the fact that every derivation $\mathcal{D}$ in $\mathbf{HB_0^{=}}$ without open assumptions can effectively be transformed into a derivation $\mathcal{D}'$ that mimics $\mathcal{D}$ and that fulfills the hypotheses (i)–(iii) of Theorem C.11 and fulfills the condition **ADA**, Theorem C.11 has the following obvious corollary.

Corollary C.14. *Every derivation $\mathcal{D}$ in $\mathbf{HB_0^{=}}$ without open assumptions and with conclusion $\tau = \sigma$, for some $\tau, \sigma \in \mu Tp$, can effectively be transformed into a derivation $\mathcal{D}'$ in $\mathbf{HB_0^{=}}$ without open assumptions and with the same conclusion as $\mathcal{D}$ such that $\mathcal{D}'$ fulfills the condition* **ADA** *and such that its depth $|\mathcal{D}'|$ is bounded by*

$$|\mathcal{D}'| < \ 2\left((s(\tau)+1)(s(\sigma)+1)+2|\tau|+2|\sigma|+2\right)\,. \tag{C.34}$$

Proof. The corollary is an immediate consequence of Lemma C.12 and Theorem C.11. □

The *height* of a formula occurrence in a derivation $\mathcal{D}$ means the number of rule applications in $\mathcal{D}$ below this occurrence, i.e. the number of rule applications passed on the thread in $\mathcal{D}$ from the considered formula occurrence downwards to the conclusion of $\mathcal{D}$.

Corollary C.15. (Elimination of open assumptions located "high enough" in $\mathbf{HB_0^{=}}$-derivations). *Let $\mathcal{D}$ be a derivation in $\mathbf{HB_0^{=}}$ with conclusion $\tau = \sigma$, for some $\tau, \sigma \in \mu Tp$, and with the properties that*

- *$\mathcal{D}$ fulfills the conditions (i) and (ii) in Theorem C.11, and that*
- *$\mathcal{D}$ contains occurrences of open marked assumptions only at heights greater than or equal to $h_{(\tau,\sigma)}$ which is defined as*

$$h_{(\tau,\sigma)} \ =_{def} 2\left((s(\tau)+1)(s(\sigma)+1)+2|\tau|+2|\sigma|+2\right)\,. \tag{C.35}$$

Then $\mathcal{D}$ can effectively be transformed into a derivation $\mathcal{D}'$ in $\mathbf{HB_0^{=}}$ without open assumptions *that mimics $\mathcal{D}$ and that has depth $|\mathcal{D}'| < h_{(\tau,\sigma)}$.*

Proof. We show the corollary by induction on the number $\#V_{\mathbf{AA}}(\mathcal{D})$ of violations of the condition **AA** in $\mathbf{HB_0^{=}}$-derivations for which the assumption of the corollary is true.

For the base case of the induction, let $\mathcal{D}$ be a derivation in $\mathbf{HB_0^{=}}$ with conclusion $\tau = \sigma$, for some $\tau, \sigma \in \mu Tp$, such that $\mathcal{D}$ fulfills the conditions (i) and (ii) in Theorem C.11, such that all open assumptions of $\mathcal{D}$ (if any) are located at heights

$\geq h_{(\tau,\sigma)}$ with $h_{(\tau,\sigma)}$ as in (C.35), and such that $\#V_{\mathbf{AA}}(\mathcal{D}) = 0$. Then Theorem C.11 can be applied and it gives $|\mathcal{D}| < h_{(\tau,\sigma)}$. Hence, by what we have supposed about its open marked assumptions, $\mathcal{D}$ cannot have open assumptions.

For the induction step, let $\mathcal{D}$ be a derivation in $\mathbf{HB}_{\mathbf{0}}^{=}$ with conclusion $\tau = \sigma$, for some $\tau, \sigma \in \mu Tp$, such that $\mathcal{D}$ fulfills the conditions (i) and (ii) in Theorem C.11, such that all open assumptions of $\mathcal{D}$ (if there are any) are located at heights $\geq h_{(\tau,\sigma)}$ with $h_{(\tau,\sigma)}$ as in (C.35), and such that $\#V_{\mathbf{AA}}(\mathcal{D}) = n+1$ for some $n \in \omega$. Then there is at least one violation of the condition **AA** in $\mathcal{D}$, i.e. an occurrence of a formula $\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2$ in $\mathcal{D}$, for some $\tau_1, \tau_2, \sigma_1, \sigma_2 \in \mu Tp$, with respect to which $\mathcal{D}$ can be represented as a symbolic prooftree of the form

$$\begin{array}{cl} \mathcal{D}_2 & \\ (\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2) & \\ \mathcal{DC}_1 & \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots & \text{ARROW or ARROW/FIX, } u \\ (\tau_1' \to \tau_2' = \sigma_1' \to \sigma_2') & \\ \mathcal{DC}_0 & \\ \tau = \sigma & \end{array}$$

with $\tau_1', \tau_2', \sigma_1', \sigma_2' \in \mu Tp$ such that $\tau_i' \equiv_{\text{ren}} \tau_i$ and $\sigma_i' \equiv_{\text{ren}} \sigma_i$ for $i \in \{1, 2\}$, with derivation contexts $\mathcal{DC}_0$ and $\mathcal{DC}_1$, and with a subderivation $\mathcal{D}_2$ of $\mathcal{D}$ that contains at least one application of a rule different from REN.

By a simplification as explained in already Remark C.1, $\mathcal{D}$ can be transformed into a derivation $\tilde{\mathcal{D}}$ of the form

$$\begin{array}{cl} \dfrac{(\tau_1' \to \tau_2' = \sigma_1' \to \sigma_2')^v}{(\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2)} & \text{REN} \\ \mathcal{DC}_1^{[[v/u]]} & \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots & \text{ARROW/FIX, } v \\ (\tau_1' \to \tau_2' = \sigma_1' \to \sigma_2') & \\ \mathcal{DC}_0 & \\ \tau = \sigma & \end{array}$$

where v is a new assumption marker (i.e. it does not occur in $\mathcal{D}$) and where $\mathcal{DC}_1^{[[v/u]]}$ is the result of changing the bottommost rule application into an application of ARROW/FIX at which assumptions marked by v are discharged, and by changing, if the bottommost rule application in $\mathcal{D}$ is an application of ARROW/FIX at which assumptions marked by u are discharged, all open marked assumption $(\tau_1' \to \tau_2' = \sigma_1' \to \sigma_2')^u$ in $\mathcal{DC}_1$ into open marked assumptions $(\tau_1' \to \tau_2' = \sigma_1' \to \sigma_2')^v$. Then $\tilde{\mathcal{D}}$ mimics $\mathcal{D}$, and it has again the property that open marked assumptions in $\tilde{\mathcal{D}}$ occur only at heights $> h_{(\tau,\sigma)}$. Furthermore $\#V_{\mathbf{AA}}(\tilde{\mathcal{D}}) < \#V_{\mathbf{AA}}(\mathcal{D})$ holds because at least one violation of **AA** has been eliminated and no other violations have been introduced. Therefore the induction hypothesis can be applied to $\tilde{\mathcal{D}}$ and it guarantees that $\tilde{\mathcal{D}}$ can effectively be transformed into a derivation $\mathcal{D}'$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ without open assumptions that mimics $\tilde{\mathcal{D}}$ and that has depth $|\mathcal{D}'| \leq h_{(\tau,\sigma)}$. Since the mimicking relation $\precsim$ is transitive, $\mathcal{D}'$ also mimics $\mathcal{D}$ and hence we have shown that $\mathcal{D}$ can be transformed into a derivation $\mathcal{D}'$ in $\mathbf{HB}^{=}_{\mathbf{0}}$ with the desired properties. $\square$

Bibliography

[AmCa93] Amadio, R.M., Cardelli, L.: "Subtyping Recursive Types", *ACM Transactions on Programming Languages and Systems 15 (4)*, pp. 575–631, 1993; we refer to page numbers of the version that is available at `http://research.microsoft.com/Users/luca/Papers/SRT.pdf` .

[And65] Anderaa, S.: "On the algebra of regular expressions", *Applied Mathematics*, Harvard University, Cambridge, January 1965, pp. 1–18.

[ArKl95] Ariola, Z.M., Klop, J.W.: "Equational Term Graph Rewriting", *Fundamenta Informaticae 26 (3,4)*, pp. 207–240, June 1996; extended version as *University of Oregon Technical Report CIS–TR–95–16*, 1995, and as *Vrije Universiteit Amsterdam Technical Report IR–391*, September 1995.

[Avr91] Avron, A.: "Simple Consequence Relations", *Information and Computation 92 (1)*, pp. 105–139, 1991.

[Ba81] Barendregt, H.P.: *The Lambda Calculus*, Elsevier, 1981, revised edition 1984.

[Br97] Brandt, M.: "Recursive Subtyping: Axiomatizations and Computational Interpretations", Master's Thesis, DIKU, Department of Computer Science, University of Copenhagen, August 1997; available as *TOPPS Technical Report D-352* at `ftp://ftp.diku.dk/diku/semantics/papers/D-352.ps.gz` .

[BrHe98] Brandt, M., Henglein, F.: "Coinductive Axiomatization of Recursive Type Equality and Subtyping", *Fundamenta Informaticae 33*, pp. 1–30, 1998.

[CaCo91] Cardone, F., Coppo, M.: "Type Inference with Recursive Types: Syntax and Semantics", *Information and Computation 92*, pp. 48–80, 1991.

[Con71] Conway, J.H.: *Regular Algebra and Finite Machines*, Chapman and Hall, London, 1971.

[Cur63] Curry, H.B.: *Foundations of Mathematical Logic*, McGraw-Hill Book Company, New York San Francisco Toronto London, 1963.

[Fit90] Fitting, M.: *First-Order Logic and Automated Theorem Proving*, Springer-Verlag New York, Berlin, Heidelberg, 1990.

[Gen35] Gentzen, G.: "Untersuchungen über das logische Schließen I, II", *Mathematische Zeitschrift 39*, S. 176–210, 405–431.

[Gra01] Grabmayer, C.: "Proof-Theoretic Interconnections between Known Inference- and Axiom-Systems for the Equality and Subtype Relations on Recursive Types", *slides* for a talk given at the *KUN Lambda Seminar*, Nijmegen, January 12, 2001; presently available from `http://www.cs.vu.nl/~clemens/ Nijmegen.ps.gz` .

[Gra02a] Grabmayer, C.: "Proving Equality for Recursive Types – A Duality between 'syntactic-matching' tableaux and Brandt-Henglein derivations", *slides* for a talk given at the *Process Algebra Meeting (PAM)*, CWI Amsterdam, May 8, 2002; presently available from `http://www.cs.vu.nl/~clemens/PAM.ps.gz` .

[Gra02b] Grabmayer, C.: "A Duality in Proof Systems for Recursive Type Equality and Bisimulation Equivalence on Cyclic Term Graphs", Proceedings of the Workshop TERMGRAPH 2002, Barcelona, October 2002, *Electronic Notes in Computer Science*, Vol. 72 No. 1, to become available at `http://www.sciencedirect.com/science/journal/15710661` .

[Gra02c] Grabmayer, C.: "A Duality in Proof Systems for Recursive Type Equality and for Bisimulation Equivalence on Cyclic Term Graphs", *Vrije Universiteit Amsterdam Rapport*, Divisie Wiskunde & Informatica, Faculteit der Exacte Wetenschappen, Rapportnr IR-499, juli 2002.

[Gra03a] Grabmayer, C.: "Derivability and Admissibility of Inference Rules in Abstract Hilbert systems", *VU Amsterdam report*, July 2003; presently available from `http://www.cs.vu.nl/~clemens/dairahs.ps` .

[Gra03b] Grabmayer, C.: "Derivability and Admissibility of Inference Rules in Abstract Hilbert systems", *Selected Poster* presented at the conference Computer Science Logic and 8th Kurt Gödel Colloquium 2003, 25–30th August 2003, Vienna, Austria.

[Gra04a] Grabmayer, C.: "Transforming proofs by coinduction into 'traditional' proofs", slides for a talk given in the *Seminar Theoretical Computer Science*, Vrije Universiteit Amsterdam, 23 January 2004, available from `http://www.cs.vu.nl/~clemens/seminar230104_slides.pdf` .

[Gra04b] Grabmayer, C.: "Derivability and Admissibility of Inference Rules in Abstract Hilbert Systems", *Extended Poster* (overview article of the poster [Gra03b]), *Collegium Logicum – Volume VIII*, Kurt Gödel Society, Vienna, 2004; available at `http://www.cs.vu.nl/~clemens/ dairahs_extposter.ps` .

[HS86] Hindley, J.R., Seldin, J.P.: *Introduction to Combinators and Lambda-calculus*, Cambridge University Press, 1986.

[Iem01] Iemhoff, R.: *Provability Logic and Admissible Rules*, PhD-thesis, Universiteit van Amsterdam, 2001; available at `http://www.logic.at/people/iemhoff/proeve.ps` .

[Klee52] Kleene, S.C.: *Introduction to Metamathematics*, North-Holland Publishing Co., Amsterdam, Groningen, 1952.

[Koz94] Kozen, D.: "A completeness theorem for Kleene algebras and the algebra of regular events", *Information and Computation 110 (2)*, pp. 366-390, May 1993.

[KPS95] Kozen, D., Palsberg, J., Schwartzbach, M.I.: "Efficient Recursive Subtyping", *Mathematical Structures in Computer Science 5*, pp. 113-125, 1995.

[Len99] Lenisa, M.: "A Complete Coinductive Logical System for Bisimulation Equivalence of Circular Objects", *FoSSaCS'99 (ETAPS) Conference Proceedings*, W. Thomas ed., Springer LNCS vol. 1578, 1999.

[Lor69] Lorenzen, P.: *Einführung in die operative Logik und Mathematik*, (Erste Auflage 1955), Zweite Auflage, Springer-Verlag, Berlin Heidelberg New York, 1969.

[Me64] Mendelson, E.: *Introduction to Mathematical Logic*, Van Nostrand & Co, 1964; the references are to the Third Edition, Wadsworth & Brooks/Cole Advanced Books & Software, Pacific Grove, California, 1987.

[Men86] Mendler, N.P.: "First and second-order lambda calculi with recursive types", Cornell University, *unpublished*.

[Mil84] Milner, R.: "A Complete Inference System for a Class of Regular Behaviours", *Journal of Computer and System Sciences 28*, pp. 439–466, 1984.

[MPS86] MacQueen, D., Plotkin, G.D., Sethi, R.: "An ideal model for recursive polymorphic types", *Information and Control 71*, pp. 95–130, 1986.

[Plu93] Plump, D.: *Evaluation of Functional Expressions by Hypergraph Rewriting*, PhD-Thesis, Universität Bremen, 1993.

[Pra65] Prawitz, D.: *Natural Deduction* (A Proof-Theoretical Study), Almqvist & Wiksell, Stockholm, Göteborg, Uppsala, 1965.

[Rut98] Rutten, J.J.M.M.: "Automata and Coinduction (an Exercise in Coinduction)", Technical Report SEN-R9803, CWI, Amsterdam, 1998; also in: Proceedings of CONCUR '98, D. Sangiorigi and R. de Simone (eds.), LNCS 1466, Springer, 1998, pp. 194–218.

[Sal66] Salomaa, A.: "Two complete axiom systems for the algebra of regular events", *Journal of the Association for Computing Machinery (JACM) 13 (1)*, pp. 158–169, 1966.

[Schm60] Schmidt, H.A.: *Mathematische Gesetze der Logik*, Vorlesungen über Aussagenlogik I, Springer Verlag, Berlin Göttingen Heidelberg, 1960.

[Schu60] Schütte, K.: *Beweistheorie*, Springer Verlag, Berlin Göttingen Heidelberg, 1960.

[Sco74] Scott, D.: "Rules and Derived Rules", in Stenlund, S.: *Logical Theory and Semantic Analysis*, 147–161, D. Reidel Publishing Company, Dordrecht–Holland, 1974.

[Shoe67] Shoenfield, J.R.: *Mathematical Logic*, Addison-Wesley Publishing Company, Reading, Mass. – Menlo Park, Cal. – London – Don Mills, Ont., 1967, 1973.

[Sm68] Smullyan, R.M.: *First Order Logic*, Springer Verlag, Berlin Heidelberg New York, 1968. Unabridged, corrected republication by: Dover Publications Inc., New York, 1995.

[Ter03] Terese: *Term Rewriting Systems*, Cambridge Tracts in Theoretical Computer Science 55, Cambridge University Press, 2003.

[Tr73] Troelstra, A.S. (ed.): *Metamathematical Investigations of Intuitionistic Arithmetic and Analysis*, Lecture-notes in Mathematics 344, Springer Verlag, Berlin, 1973.

[TS00] Troelstra, A.S., Schwichtenberg, H.: *Basic Proof Theory*, Cambridge Tracts in Theoretical Computer Science, Cambridge University Press, 1996; 2nd, revised edition, 2000 (specific references are directed to this edition).

[VB01] Vestergaard, R., Brotherston, J.: "A Formalised First-Order Confluence Proof for the λ-Calculus Using One-Sorted Variable Names (Barendregt was right after all … almost)", pp. 306–321 in Middeldorp, A. (ed.): "Rewriting Techniques and Applications", *Proceedings of RTA 2001*, LNCS 2051, Springer-Verlag, Berlin Heidelberg New York, 2001.

[vOdV02] van Oostrom, V., de Vrijer, R.C.: "Four equivalent equivalences of reductions", *Artificial Intelligence Preprint Series*, Preprint nr. 035, Onderwijsinstituut CKI, Utrecht University, December 2002; this report can be found via `http://preprints.phil.uu.nl/aips/` .

Index of Notations

Subject Index

Samenvatting

De titel van dit proefschrift luidt, in het Nederlands vertaald: “Het Relateren van Bewijssystemen voor Recursieve Types.” Het hierdoor aangeduide onderwerp kan als volgt nader worden omschreven: er wordt de vraagstelling onderzocht op welke manier verschillende, uit de literatuur bekende, bewijssystemen voor de relatie “gelijkheid van recursieve types” door effectieve, bewijstheoretische transformaties met elkaar in verband gebracht kunnen worden.

Als “types” worden in de informatica gewoonlijk zodanige collecties van waarden beschouwd die met elkaar een bepaalde structuur of vorm gemeen hebben. Zo komen in de meeste computertalen “fundamentele types” als Int, Real, en $\mathsf{Boolean}$ voor, die respectievelijk voor de gehele getallen, voor de reële getallen, en voor de twee waarheidswaarden “true” en “false” staan. Dat geldt ook voor “samengestelde types” als bijvoorbeeld $\mathsf{Int} \times \mathsf{Int}$, $\mathsf{Real} \to \mathsf{Boolean}$, en $\mathsf{Int} + (\mathsf{Real} \times \mathsf{Real})$, die de verzamelingen van paren van gehele getallen, van functies op de reële getallen met waarden in $\mathsf{Boolean}$ (*Boolean functions*), en van de disjuncte vereniging van de gehele getallen en van paren van reële getallen symboliseren. Hierbij zijn $\times$, $\to$, en $+$ zogeheten “type-constructoren”, dat wil zeggen operatoren die geschikt zijn om nieuwe types met gebruik van al bestaande types te vormen.

Ook “recursieve types” worden in vele computertalen gebruikt. Ze voldoen aan recursieve vergelijkingen; zo voldoet bijvoorbeeld het type List van eindige of oneindige lijsten van gehele getallen aan de vergelijking

$$\mathsf{List} = \mathsf{Unit} + (\mathsf{Int} \times \mathsf{List}) \tag{1}$$

waarbij Unit een fundamenteel type is met maar één waarde. Onder bepaalde omstandigheden is het mogelijk dat een recursieve type door één vergelijking op eenduidige wijze is gedefinieerd. Het ligt dan voor de hand om *de* oplossing van een vergelijking door een speciale term aan te duiden. In het geval van (1) zou dat de term $\mu\alpha.\,(\mathsf{Unit} + (\mathsf{Int} \times \alpha))$ kunnen zijn; in deze “μ-term” wordt het vrije voorkomen van α in $\mathsf{Unit} + (\mathsf{Int} \times \alpha)$ door de μ-binder aan het begin gebonden. In het algemeen kan de μ-term $\mu\alpha.\,\tau(\alpha)$ voor *de* oplossing van een recursieve vergelijking $\sigma = \tau(\sigma)$ worden gebruikt, als deze inderdaad uniek oplosbaar is (met gebruikmaking van een expliciete substitutie-operatie kan de vergelijking $\sigma = \tau(\sigma)$ ook als $\sigma = \tau[\sigma/\alpha]$ worden geschreven).

Om formeel te kunnen rekenen met μ-termen die oplossingen van recursieve vergelijkingen symboliseren, zijn de axioma's

$$\mu\alpha.\,\tau = \tau[\mu\alpha.\,\tau/\alpha] \qquad \text{(waarbij } \tau \text{ een } \mu\text{-term, } \alpha \text{ en type-variabele)} \tag{2}$$

noodzakelijk. In een stap vanuit een term $\mu\alpha.\,\tau$ naar een term $\tau[\mu\alpha.\,\tau/\alpha]$ (de uitkomst van de substitutie van $\mu\alpha.\,\tau$ voor alle vrije voorkomens van α in τ) wordt $\mu\alpha.\,\tau$ "ontvouwen", terwijl in een overgang in de andere richting $\tau[\mu\alpha.\,\tau/\alpha]$ "gevouwen" wordt met de uitkomst $\mu\alpha.\,\tau$. Omdat het mogelijk is te laten zien dat (1) een unieke oplossing heeft, is er reden om de oplossing van (1) inderdaad door een μ-term aan te geven: we laten τ_1 gedefinieerd door

$$\tau_1 \equiv_{\text{def}} \mu\alpha.\,(\mathsf{Unit} + (\mathsf{Int} \times \alpha)) \ .$$

Door gebruik van een geschikt axioma uit het schema (2) is het nu mogelijk om ook formeel aan te tonen dat de μ-term τ_1 aan de vergelijking (1) voldoet als hij daar voor List wordt gesubstitueerd:

$$\tau_1 \equiv \mu\alpha.\,(\mathsf{Unit} + (\mathsf{Int} \times \alpha)) = (\mathsf{Unit} + (\mathsf{Int} \times \alpha))[\tau_1/\alpha] \equiv \mathsf{Unit} + (\mathsf{Int} \times \tau_1) \ .$$

Types die aan recursieve vergelijkingen voldoen, vervullen meestal oneindig veel vergelijkingen. Daarom kan het gebeuren dat twee niet-identieke μ-termen dezelfde recursieve typen symboliseren. Het recursieve type List voldoet bijvoorbeeld ook aan de vergelijking

$$\mathsf{List} = \mathsf{Unit} + (\mathsf{Int} \times (\mathsf{Unit} + (\mathsf{Int} \times \mathsf{List}))) \ .$$

Omdat deze vergelijking wederom eenduidig oplosbaar is, staat ook de μ-term

$$\tau_2 \equiv_{\text{def}} \mu\alpha.\,(\mathsf{Unit} + (\mathsf{Int} \times (\mathsf{Unit} + (\mathsf{Int} \times \alpha))))$$

voor het type List, evenzo als τ_1. Voorbeelden als dit leiden tot het begrip "gelijkheid van recursieve types" (*recursive type equality*): dat is een binaire equivalentierelatie die twee μ-term-beschrijvingen σ_1 en σ_2 van recursieve types RType_1 en RType_2 relateert dan en slechts dan als ze hetzelfde type symboliseren (en dat betekent als $\mathsf{RType}_1 = \mathsf{RType}_2$). De benaming van deze relatie, die strict genomen "gelijkheid van gesymboliseerde recursieve typen" moest zijn, komt voort uit het feit dat in de literatuur ook μ-termen die voor recursieve types staan, "recursieve types" worden genoemd; dat gebeurt in het volgende hier ook.

In dit proefschrift worden bewijssystemen bestudeerd die correct en volledig zijn voor de relatie "gelijkheid van recursieve types". Maar er wordt alleen een beperkte klasse van recursieve types[4] bekeken: types die door een grammatica als

$$\tau \ ::= \ \bot \mid \top \mid \alpha \mid \tau \to \tau \mid \mu\alpha.\,\tau$$

[4] De hier gedefinieerde recursieve types zijn eigenlijk μ-term-beschrijvingen van recursieve types (en dus beschrijvingen van types die verzamelingen van waarden zijn welke aan recursieve vergelijkingen voldoen). Maar, zoals eerder opgemerkt, de naam "recursieve type" wordt in de literatuur overwegend ook voor μ-term-beschrijvingen van recursieve types gebruikt.

(waarbij $\bot$, $\top$ type-constanten, en α type-variabelen) worden gegenereerd. Hierbij horen dus precies alle recursieve types die uit twee type-constanten $\bot$ en $\top$, en een gegeven verzameling van type-variabelen met behulp van $\to$ als de enige type-constructor gevormd kunnen worden.

Er worden drie bewijssystemen (en een aantal van nieuw geïntroduceerde variant-systemen) voor de relatie "gelijkheid van recursieve types" bestudeerd: twee axiomatische afleidingssystemen in gebruikelijke zin, en een bewijssysteem dat strict genomen geen axiomatisch afleidingssysteem is: het is bedacht om tests te kunnen uitvoeren naar de consistentie met het systeem van gegeven vergelijkingen tussen recursieve types. De axiomatische afleidingssystemen zijn geïntroduceerd door Amadio and Cardelli (1993), en door Brandt en Henglein (1998). Het eerste systeem is van een "traditionele" soort, aangezien het in een rij van reeds eerder bekende, vergelijkbare bewijssystemen voor andere doeleinden geplaatst kan worden. Daarentegen is het tweede systeem van nieuwere vorm; het kan door methodes uit de coalgebra gefundeerd worden. Het derde systeem is afkomstig uit werk van Ariola en Klop (1995).

Het systeem van Amadio en Cardelli wordt vooral gekenmerkt door de aanwezigheid van een regel UFP (*unique fixed-point* rule):

$$\frac{\tau_1 = \tau[\tau_1/\alpha] \qquad \tau_2 = \tau[\tau_2/\alpha]}{\tau_1 = \tau_2}\ \text{UFP} \quad (\text{if } \alpha \text{ is } \textit{guarded} \text{ in } \tau)$$

Met deze regel kunnen twee recursieve types gelijk worden bewezen als ze dezelfde *guarded* recursieve vergelijking vervullen.

Het bewijssysteem van Brandt en Henglein bevat een, op het eerste gezicht, paradoxale afleidingsregel die circulair lijkt. Dit is de regel ARROW/FIX die, in een formalisering die geschikt is voor een systeem van natuurlijke deductie, toepassingen van de vorm

$$\frac{\begin{array}{c}[\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2]^u \\ \mathcal{D}_1 \\ \tau_1 = \sigma_1\end{array} \qquad \begin{array}{c}[\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2]^u \\ \mathcal{D}_2 \\ \tau_2 = \sigma_2\end{array}}{\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2}\ \text{ARROW/FIX}, u$$

heeft. Deze regel staat toe, dat in een afleiding $\mathcal{D}$, zoals boven afgebeeld, van een vergelijking $\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2$ dezelfde vergelijking ook een of meerdere keren als assumptie mag worden gebruikt, zonder dat de afleiding $\mathcal{D}$ in haar geheel uiteindelijk nog van de gebruikte assumpties van de vorm $\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2$ afhankelijk blijft; deze assumpties worden namelijk door de toepassing van ARROW/FIX beneden in $\mathcal{D}$ "ingetrokken". Deze afleidingsregel houdt verband met de coalgebraïsche motivatie voor het bewijssysteem van Brandt en Henglein. Dat heeft het gevolg dat de correctheid van het system van Brandt en Henglein ten opzichte van de relatie "gelijkheid van recursieve types" niet vanzelfsprekend is; meestal wordt de correctheid van het systeem dan ook door een technisch ingewikkeld argument bewezen.

De regel ARROW/FIX in het systeem van Brandt en Henglein kan als een bijzondere vorm van een "compositie-regel" ARROW worden beschouwd omdat de conclusie $\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2$ van een toepassing van deze regel uit de premissen $\tau_1 = \sigma_1$ en $\tau_2 = \sigma_2$ wordt "gecomponeerd". Daarentegen bevat het systeem van Ariola en Klop (preciezer gezegd, de hier bekeken adaptie voor recursieve types van dat systeem) een "decompositie-regel" met toepassingen van de vorm

$$\frac{\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2}{\tau_i = \sigma_i} \quad (\text{voor elke } i \in \{1,2\})$$

die een vergelijking tussen twee samengestelde recursieve type op twee mogelijke manieren uit elkaar halen, oftewel "decomponeren", kan.

Het uitgangspunt voor dit proefschrift was een door Klop gedane observatie die het aannemelijk maakte dat er een direct verband bestaat tussen de volgende twee activiteiten: (a) het aantonen dat een vergelijking $\tau = \sigma$ tussen recursieve types τ en σ consistent is met het bewijssysteem van Ariola en Klop; en (b) het afleiden van de vergelijking $\tau = \sigma$ in het systeem van Brandt en Henglein. Het streven om deze observatie formeel aan te tonen, werd ook daardoor gemotiveerd dat ze een nieuwe mogelijkheid suggereerde om de correctheid van het systeem van Brandt en Henglein te bewijzen. Hiernaast deden zich daarenboven nog een aantal verband houdende vragen voor: Zouden misschien nog andere opmerkelijke bewijstheoretische connecties bestaan tussen de systemen van Amadio en Cardelli, Brandt en Henglein, en Ariola en Klop? Zou het bijvoorbeeld mogelijk zijn, om een "circulaire", coïnductive redenering, geformaliseerd als een afleiding in het systeem van Brandt en Henglein, om te zetten naar een derivatie, een formeel bewijs, in het systeem van Amadio en Cardelli? En zou ook andersom een afleiding in het systeem van Amadio en Cardelli altijd op effectieve manier getransformeerd kunnen worden naar een afleiding in het systeem van Brandt en Henglein?

Deze vragen zijn in het proefschrift onderzocht en hebben uiteindelijk direct tot de Hoofdstukken 6–8 daarvan geleid. In elk van deze drie hoofdstukken wordt een effectieve bewijstheoretische transformatie beschreven die in staat is om een willekeurige afleiding $\mathcal{D}_2$ van een theorema A in een systeem $\mathcal{S}_1$ op een stapsgewijze manier te kunnen omvormen tot een afleiding $\mathcal{D}_2$ in $\mathcal{S}_2$ die aantoont dat A ook een theorema van $\mathcal{S}_2$ is; hierbij zijn $\mathcal{S}_1$ en $\mathcal{S}_2$ telkens bewijssystemen van een van de boven besproken drie soorten. In Hoofdstuk 6 wordt de eerder genoemde observatie van Klop concreet gemaakt door het definiëren van twee transformaties die het bestaan van een "dualiteit" laten zien tussen het systeem van Ariola en Klop en van het systeem van Brandt en Henglein. In Hoofdstuk 7 wordt een transformatie aangegeven van afleidingen in het systeem van Amadio en Cardelli naar afleidingen in het systeem van Brandt en Henglein; en uiteindelijk wordt in Hoofdstuk 8 ook een transformatie in de andere richting ontwikkeld en uitgelegd: een van afleidingen in het systeem van Brandt en Henglein naar afleidingen in het systeem van Amadio en Cardelli. Deze transformaties vormen, samen met sommigen van hun onderdelen, het voornaamste resultaat: een "netwerk" van transformaties dat alle behandelde systemen met elkaar verbindt. Dit netwerk wordt in Sectie 9.1 van Hoofdstuk 9 in de vorm van een afbeelding aanschouwelijk gemaakt, en vervolgens besproken.

Hieronder wordt de inhoud van de belangrijkste hoofdstukken van het proefschrift in meer detail beschreven.

In Hoofdstuk 3 worden de meest fundamentele begrippen in verband met de hier behandelde klasse van recursieve types geïntroduceerd. Hierbij horen eerst de subklasse van recursieve types *in canonical form*, de notie van "substitutie expressies" voor recursieve types, en een formele definitie van toelaatbare herbenoemingen van gebonden variabelen in een recursieve type (de *variant relation* tussen recursieve types, in analogie met α-conversie in λ-calculus). En verder het belangrijke begrip van de "boom-ontvouwing" (*tree unfolding*) van een recursieve type, waardoor aan recursieve types uit de gedefinieerde klasse een semantiek wordt toegekend: ieder recursief type τ kan als eindige beschrijving van zijn "boom-ontvouwing" $\mathbf{Tree}(\tau)$, een oneindige binaire boom, worden opgevat. Deze notie leidt vervolgens rechtstreeks naar de hier centraal staande binaire relatie $=_\mu$, genoemd "gelijkheid van recursieve types", die als volgt is gedefinieerd: voor twee recursieve types τ en σ geldt $\tau =_\mu \sigma$ dan en slechts dan als $\mathbf{Tree}(\tau) = \mathbf{Tree}(\sigma)$, en dus dan en slechts dan als τ en σ dezelfde boom-ontvouwing bezitten.

In Hoofdstuk 4 worden de noties "afleidbaarheid" en "toelaatbaarheid" van deductieregels in formele bewijssystemen behandeld. Als motivatie van dit hoofdstuk binnen dit proefschrift wordt in Sectie 4.1 eerst uitgelegd waarom deze noties belangrijk zijn voor het vinden van bewijstheoretische transformaties in het algemeen. In Sectie 4.2 en in Sectie 4.3 worden de noties afleidbaarheid en toelaatbaarheid van deductieregels geïntroduceerd. Dat gebeurd apart in de twee klassen van bewijssystemen waarbij alle systemen horen die in latere hoofdstukken voorkomen: in "pure" Hilbert-stijl afleidingssystemen, en in natuurlijke-deductie systemen. De noties "afleidbaarheid" en "toelaatbaarheid" van deductieregels worden in Appendix B in abstracte versies van bewijssystemen preciezer uitgewerkt. Hiervoor worden de concepten *abstract pure Hilbert systems* en *abstract natural-deduction systems* gedefinieerd.

In Hoofdstuk 5 worden de drie reeds eerder genoemde bewijssystemen voor "gelijkheid van recursieve types" op een precieze en formele manier geïntroduceerd: enerzijds, in Sectie 5.1, de axiomatische systemen $\mathbf{AC}^{=}$ van Amadio en Cardelli, en $\mathbf{HB}^{=}$ van Brandt en Henglein, en anderzijds, in Sectie 5.2, het uit werk van Ariola en Klop voortkomende bewijssysteem $\mathbf{AK}^{=}$. Voor deze systemen worden verder nog bepaalde varianten gedefinieerd die later van pas zullen komen. Hierbij horen de versies $\mathbf{HB}_\mathbf{0}^{=}$ van $\mathbf{HB}^{=}$ en $\mathbf{AK}_\mathbf{0}^{=}$ van $\mathbf{AK}^{=}$, die sterkere bewijstheoretische eigenschappen hebben dan de oorspronkelijke systemen. In tegenstelling met $\mathbf{HB}^{=}$ en $\mathbf{AK}^{=}$ zijn de systemen $\mathbf{HB}_\mathbf{0}^{=}$ en $\mathbf{AK}_\mathbf{0}^{=}$ namelijk "analytisch": deducties in $\mathbf{HB}_\mathbf{0}^{=}$ en in $\mathbf{AK}_\mathbf{0}^{=}$ voldoen aan een passend bij het respectievelijke systeem geformuleerde "deelformule-eigenschap".[5] De variant-systemen $\mathbf{HB}_\mathbf{0}^{=}$ en $\mathbf{AK}_\mathbf{0}^{=}$ tonen zich in de

[5] Gewoonlijk wordt in de bewijstheorie over een bewijssysteem $\mathcal{S}$ gezegd dat $\mathcal{S}$ de "deelformule-eigenschap" (*subformula property*) heeft, of dat het "deelformule-principe" (*subformula principle*) in $\mathcal{S}$ geldig is, als alle afleidingen $\mathcal{D}$ in $\mathcal{S}$ de "deelformule-eigenschap" hebben: alle in $\mathcal{D}$ voorkomende formules zijn ook in de conclusie van $\mathcal{D}$ als deelformules bevat. Het geldig zijn van dit principe is een zeer wenselijk attribuut van een bewijssysteem $\mathcal{S}$, omdat daardoor enerzijds het zoeken naar concrete deducties in $\mathcal{S}$, en anderzijds het bewijzen van meta-resultaten over bewijsbaarheid in $\mathcal{S}$ beduidend wordt vergemakkelijkt.

Hoofdstukken 6–8 zeer geschikte hulpmiddelen voor het vinden van bewijstheoretische transformaties tussen de systemen $\mathbf{AC}^{=}$, $\mathbf{HB}^{=}$, en $\mathbf{AK}^{=}$. Uiteindelijk worden in Sectie 5.3 van Hoofdstuk 5 ook een aantal fundamentele verschillen bewezen tussen de bewijstheoretische eigenschappen van de hier gedefinieerde axiomatische en de “syntactic-matching” systemen.

Het onderwerp in Hoofdstuk 6 is het door de observatie van Klop aannemelijk gemaakte verband tussen het vinden van een afleiding in het systeem $\mathbf{HB}^{=}$ van Brandt en Henglein en het bewijzen van consistentie in het system $\mathbf{AK}^{=}$ van Ariola en Klop. Hier wordt aangetoond dat deze twee activiteiten in feite duaal zijn als ze op de analytische versies $\mathbf{HB}_{\mathbf{0}}^{=}$ en $\mathbf{AK}_{\mathbf{0}}^{=}$ van $\mathbf{HB}^{=}$ en $\mathbf{AK}^{=}$ worden betrokken (voor de systemen $\mathbf{HB}^{=}$ en $\mathbf{AK}^{=}$ bestaat een vergelijkbaar direct verband namelijk niet in volle algemeenheid). De belangrijkste drie stappen op de weg naar het bewijs hiervan zijn de volgende: (1) De introductie van een conservatieve extensie $\mathbf{e\text{-}HB}_{\mathbf{0}}^{=}$ van het systeem $\mathbf{e\text{-}HB}_{\mathbf{0}}^{=}$ die meer coïnductieve regels bevat; (2) de definitie van een special soort afleidingsbomen in $\mathbf{AK}_{\mathbf{0}}^{=}$ die “naar beneden groeien”, en daarop gebaseerd, de definitie van *consistency-unfoldings* in $\mathbf{AK}_{\mathbf{0}}^{=}$ als afleidingsbomen in $\mathbf{AK}_{\mathbf{0}}^{=}$ die als “getuigen” kunnen fungeren voor de consistentie met $\mathbf{AK}_{\mathbf{0}}^{=}$ van de formule aan hun respectievelijke wortel; (3) de definitie van een paar van geometrische gemotiveerde “spiegelingsfuncties” tussen derivaties in $\mathbf{e\text{-}HB}_{\mathbf{0}}^{=}$ en de eerder gedefinieerde afleidingsbomen in $\mathbf{AK}_{\mathbf{0}}^{=}$. De voornaamste stelling in dit hoofdstuk formuleert vervolgens het bestaan van, inderdaad, een dualiteit tussen $\mathbf{e\text{-}HB}_{\mathbf{0}}^{=}$ en $\mathbf{AK}_{\mathbf{0}}^{=}$: de spiegelingsfuncties zijn bijektief tussen de verzameling van derivaties in $\mathbf{e\text{-}HB}_{\mathbf{0}}^{=}$ zonder assumpties en de verzameling van *consistency-unfoldings* in $\mathbf{AK}_{\mathbf{0}}^{=}$. In het bijzonder kunnen derivaties in $\mathbf{e\text{-}HB}_{\mathbf{0}}^{=}$ zonder assumpties op een bijna uitsluitend geometrische manier worden “gespiegeld” met de uitkomst van een *consistency-unfolding* in $\mathbf{AK}_{\mathbf{0}}^{=}$; en andersom.

In de Hoofstukken 7 en 8 worden de systemen $\mathbf{AC}^{=}$ van Amadio/Cardelli en $\mathbf{HB}^{=}$ van Brandt en Henglein door middel van twee effectieve bewijstheoretische transformaties aan elkaar gerelateerd. Eerst wordt in Hoofdstuk 7 beschreven hoe een willekeurige derivatie $\mathcal{D}^{(\mathrm{ac})}$ in $\mathbf{AC}^{=}$ zonder assumpties stapsgewijs kan worden omgevormd met als uitkomst een derivatie $\mathcal{D}^{(\mathrm{hb})}$ in $\mathbf{HB}^{=}$ zonder open assumpties zodat $\mathcal{D}^{(\mathrm{hb})}$ dezelfde conclusie als $\mathcal{D}^{(\mathrm{ac})}$ heeft. En later wordt in Hoofdstuk 8 aangetoond dat ook andersom elke derivatie $\mathcal{D}^{(\mathrm{hb})}$ in $\mathbf{HB}^{=}$ zonder open assumpties op een effectieve manier getransformeerd kan worden in een derivatie $\mathcal{D}^{(\mathrm{ac})}$ zonder assumpties en met dezelfde conclusie als $\mathcal{D}^{(\mathrm{hb})}$.

De transformatie tussen derivaties in $\mathbf{AC}^{=}$ en derivaties in $\mathbf{HB}^{=}$ die in Hoofdstuk 7 wordt ontwikkeld en uitgelegd, bestaat in het kort uit drie stappen, die in het volgende worden beschreven. In de eerste stap worden van een gegeven derivatie $\mathcal{D}$ in $\mathbf{AC}^{=}$ zonder assumpties alle toepassingen van de zogeheten μ-compatibiliteitsregel in $\mathbf{AC}^{=}$ geëlimineerd met als resultaat een derivatie $\mathcal{D}^{(1)}$ zonder assumpties en met dezelfde conclusie als $\mathcal{D}$. Deze transformatiestap is gebaseerd op de voornaamste stelling in Sectie 7.1: deze zegt dat de μ-compatibiliteitsregel in $\mathbf{AC}^{=}$ overbodig wordt zodra aan het systeem $\mathbf{AC}^{=}$ axioma’s van de vorm $\mu\alpha\alpha_1 \ldots \alpha_n.\, \alpha = \bot$ worden toegevoegd. In de tweede stap van de transformatie wordt de derivatie $\mathcal{D}^{(1)}$ omgezet naar een derivatie $\mathcal{D}^{(2)}$ in een extensie van $\mathbf{HB}^{=}$ met als bijkomende regel een zodanige versie $\mathrm{UFP}^{-}_{(\mathrm{nd})}$ van de regel UFP in $\mathbf{AC}^{=}$ die geschikt is voor natuurlijke-deductie systemen; verder gebeurt dat zo dat $\mathcal{D}^{(2)}$ geen open assumpties bevat en dezelfde conclusie als $\mathcal{D}^{(1)}$ en als $\mathcal{D}$ heeft. In de derde en laatste stap van de transformatie worden uit de derivatie $\mathcal{D}^{(2)}$ alle voorkomens van $\mathrm{UFP}^{-}_{(\mathrm{nd})}$ geeliminieerd. Deze stap wordt gerechtvaardigd door een stelling in Sectie 7.2: de regel $\mathrm{UFP}^{-}_{(\mathrm{nd})}$ is afleidbaar in $\mathbf{HB}^{=}$. De uitkomst van stap drie, en dus van de transformatie in haar geheel, is uiteindelijk een derivatie $\mathcal{D}'$ in $\mathbf{HB}^{=}$ zonder open assumpties en met dezelfde conclusie als $\mathcal{D}$.

De in Hoofdstuk 8 ontwikkelde transformatie van derivaties in $\mathbf{HB}^{=}$ naar derivaties in $\mathbf{AC}^{=}$ bestaat uit twee componenten. Ten eerste uit een transformatie van derivaties zonder open assumpties in het variant-systeem $\mathbf{HB}^{=}_{\mathbf{0}}$ van $\mathbf{HB}^{=}$ naar derivaties met respectievelijk dezelfde conclusie en zonder assumpties in $\mathbf{AC}^{=}$; dit onderdeel wordt in Sectie 8.1 beschreven en kan als de "kern" van de gehele transformatie beschouwd worden. En ten tweede bestaat de transformatie tussen $\mathbf{HB}^{=}$ en $\mathbf{AC}^{=}$ nog uit een effectieve procedure die een derivatie zonder assumpties in het systeem $\mathbf{HB}^{=}$ van Brandt en Henglein naar een derivatie met dezelfde conclusie en zonder assumpties in het variant-systeem $\mathbf{HB}^{=}_{\mathbf{0}}$ kan omvormen; de werkwijze van deze component wordt in Sectie 8.2 uitgelegd. De in Sectie 8.1 ontwikkelde transformatie van $\mathbf{HB}^{=}_{\mathbf{0}}$-derivaties naar $\mathbf{AC}^{=}$-derivaties gebruikt als hulpmiddel een geannoteerde versie $\mathbf{ann}\text{-}\mathbf{HB}^{=}_{\mathbf{0}}$ van het bewijssysteem $\mathbf{HB}^{=}_{\mathbf{0}}$: een derivatie $\mathcal{D}$ in $\mathbf{HB}^{=}_{\mathbf{0}}$ zonder assumpties wordt namelijk eerst, door het vinden van passende annotaties (deze zijn ook recursieve types), tot een derivatie $\hat{\mathcal{D}}$ geannoteerd, voordat uit $\hat{\mathcal{D}}$ twee $\mathbf{AC}^{=}$-derivaties kunnen worden geëxtraheerd die uiteindelijk op eenvoudige manier verbonden kunnen worden tot een derivatie $(\hat{\mathcal{D}})'$ in $\mathbf{AC}^{=}$ zonder assumpties en met dezelfde conclusie als $\mathcal{D}$. De in Sectie 8.2 beschreven procedure kan als een manier voor het "normaliseren" van $\mathbf{HB}^{=}$-derivaties worden beschouwd: deze transformatie vormt een gegeven derivatie $\mathcal{D}$ zonder assumpties in $\mathbf{HB}^{=}$ eerst om naar een derivatie $\tilde{\mathcal{D}}$ zonder assumpties in het systeem $\mathbf{HB}^{=}_{\mathbf{0}}$+SYMM+TRANS, en elimineert vervolgens alle toepassingen van de regels SYMM en TRANS uit $\tilde{\mathcal{D}}$ met als uitkomst een derivatie $\mathcal{D}'$ in $\mathbf{HB}^{=}_{\mathbf{0}}$ zonder assumpties en met dezelfde conclusie als $\mathcal{D}$. De hierin gebruikte effectieve manier om toepassingen van SYMM en TRANS uit een derivatie te verwijderen, vertoond een analogie met in de bewijstheorie gebruikelijke procedures voor het elimineren van toepassingen van de regel "Cut" uit derivaties in sequenten-stijl bewijssystemen (*cut-elimination procedures*).

Het laatste hoofdstuk, Hoofdstuk 9, bestaat uit twee delen. In Sectie 9.1 wordt de samenwerking van de gevonden transformaties door een afbeelding aanschouwelijk gemaakt die de vorm van een "netwerk" heeft; ook worden frappante onderlinge relaties besproken die in deze afbeelding zichtbaar naar voren komen. En in Sectie 9.2 worden mogelijkheden geschetst hoe de bereikte resultaten veralgemeend oftewel voor soortgelijke bewijssystemen toegepast kunnen worden.