A Duality in Proof Systems for Recursive Type Equality and Bisimulation Equivalence on Cyclic Term Graphs

Clemens Grabmayer

clemens@cs.vu.nl
www.cs.vu.nl/~clemens

Department of Computer Science
Vrije Universiteit Amsterdam

October 7th, 2002
TERMGRAPH 2002
Barcelona

Overview

(I) A Duality in Proof Systems for Recursive Type Equality.

(II) A Duality in Proof Systems for Bisimulation Equivalence $\leftrightarrows$ on Cyclic Term Graphs.

Overview

(I) A Duality in Proof Systems for Recursive Type Equality.

- *Preliminaries, the basic observation.*
  - Recursive types in canonical form. Tree unfolding. Notion of 'Recursive Type Equality'.
  - The proof systems $\mathbf{HB}_{\mathbf{0}}^{=}$ and $\mathbf{AK}_{\mathbf{0}}^{=}$: These are respective variants of systems introduced by Brandt, Henglein and by Ariola, Klop.
  - *Basic observation*: A striking similarity between 'showing consistency in $\mathbf{AK}_{\mathbf{0}}^{=}$' and 'finding proofs in $\mathbf{HB}_{\mathbf{0}}^{=}$'.

(II) A Duality in Proof Systems for Bisimulation Equivalence $\leftrightarrows$ on Cyclic Term Graphs.

Overview

(I) A Duality in Proof Systems for Recursive Type Equality.

- *Preliminaries, the basic observation.*
  - Recursive types in canonical form. Tree unfolding. Notion of 'Recursive Type Equality'.
  - The proof systems $\mathbf{HB}_{\mathbf{0}}^{=}$ and $\mathbf{AK}_{\mathbf{0}}^{=}$: These are respective variants of systems introduced by Brandt, Henglein and by Ariola, Klop.
  - *Basic observation*: A striking similarity between 'showing consistency in $\mathbf{AK}_{\mathbf{0}}^{=}$' and 'finding proofs in $\mathbf{HB}_{\mathbf{0}}^{=}$'.
- *The Formalization.*
  - Step 1: The conservative extension $\mathbf{e\text{-}HB}_{\mathbf{0}}^{=}$ of $\mathbf{HB}_{\mathbf{0}}^{=}$.
  - Step 2: ('Partial') 'Consistency-Unfoldings' in $\mathbf{AK}_{\mathbf{0}}^{=}$.
  - Step 3: 'Reflection Mappings' between $\mathbf{AK}_{\mathbf{0}}^{=}$-p.c.u.'s and $\mathbf{HB}_{\mathbf{0}}^{=}$-derivations.
  - **Main Theorem:** Existence of a 'duality' between derivations in $\mathbf{HB}_{\mathbf{0}}^{=}$ and consistency-unfoldings in $\mathbf{AK}_{\mathbf{0}}^{=}$.

(II) A Duality in Proof Systems for Bisimulation Equivalence $\leftrightarrows$ on Cyclic Term Graphs.

Sketch how the result in (I) can be transferred to similar proof systems for $\leftrightarrows$.

(I)

A Duality in Proof Systems for Recursive Type Equality

Preliminaries. The considered set of recursive types.

We let $TVar$ be a countable set of *type variables*.

Definition 1. (The set $can\text{-}\mu Tp$ of recursive types in canonical form).

$$\tau ::= \alpha \mid \bot \mid \top \mid (\tau_1 \to \tau_2) \mid \mu\alpha.\underbrace{(\tau_1 \to \tau_2)}_{\text{where } \alpha \in \mathrm{fv}(\tau_1 \to \tau_2)} \qquad (\alpha \in TVar).$$

(The *leading symbol* $\mathcal{L}(\tau)$ of the recursive type τ is coloured in orange).
For arbitary $\tau, \sigma \in TVar$ we denote the assertion that τ and σ are *variants* (w.r.t. renaming of bound variables) of each other by $\tau \equiv_{\mathrm{v}} \sigma$.

$\boxtimes$

Preliminaries. The considered set of recursive types.

We let $TVar$ be a countable set of *type variables*.

Definition 1. (The set *can-μTp* of recursive types in canonical form).

$$\tau ::= \alpha \mid \bot \mid \top \mid (\tau_1 \to \tau_2) \mid \mu\alpha.\underbrace{(\tau_1 \to \tau_2)}_{\text{where } \alpha \in \mathrm{fv}(\tau_1 \to \tau_2)} \qquad (\alpha \in TVar).$$

(The *leading symbol* $\mathcal{L}(\tau)$ of the recursive type τ is coloured in orange).
For arbitary $\tau, \sigma \in TVar$ we denote the assertion that τ and σ are *variants* (w.r.t. renaming of bound variables) of each other by $\tau \equiv_{\mathrm{v}} \sigma$.

⊠

Examples: $\mu\beta.(\beta \to \bot)$, $\mu\alpha.((\alpha \to \alpha) \to \alpha)$, **but not:** $\mu\gamma.\gamma$, $\mu\alpha.(\beta \to \gamma)$.
The operations *'unfolding'* and its converse *'folding'*

$$\mu\alpha.\tau_0 \longrightarrow_{\mathrm{unfold}} \tau_0[\mu\alpha.\tau_0/\alpha] \qquad \text{and} \qquad \tau_0[\mu\alpha.\tau_0/\alpha] \longrightarrow_{\mathrm{fold}} \mu\alpha.\tau_0$$

play an important part in reasoning with and for applying recursive types.

Preliminaries. The considered set of recursive types.

We let $TVar$ be a countable set of *type variables*.

Definition 1. (The set $can\text{-}\mu Tp$ of recursive types in canonical form).

$$\tau ::= \alpha \mid \bot \mid \top \mid (\tau_1 \to \tau_2) \mid \mu\alpha.\underbrace{(\tau_1 \to \tau_2)}_{\text{where } \alpha \in \mathrm{fv}(\tau_1 \to \tau_2)} \qquad (\alpha \in TVar).$$

(The *leading symbol* $\mathcal{L}(\tau)$ of the recursive type τ is coloured in orange).
For arbitary $\tau, \sigma \in TVar$ we denote the assertion that τ and σ are *variants* (w.r.t. renaming of bound variables) of each other by $\tau \equiv_{\mathrm{v}} \sigma$.

⊠

Examples: $\mu\beta.(\beta \to \bot)$, $\mu\alpha.((\alpha \to \alpha) \to \alpha)$, **but not:** $\mu\gamma.\gamma$, $\mu\alpha.(\beta \to \gamma)$.
The operations *'unfolding'* and its converse *'folding'*

$$\mu\alpha.\tau_0 \longrightarrow_{\mathrm{unfold}} \tau_0[\mu\alpha.\tau_0/\alpha] \qquad \text{and} \qquad \tau_0[\mu\alpha.\tau_0/\alpha] \longrightarrow_{\mathrm{fold}} \mu\alpha.\tau_0$$

play an important part in reasoning with and for applying recursive types.

Definition 2. (Recursive Type Equality $=_\mu$ (Strong Recursive Type Equivalence))

For all $\tau, \sigma \in can\text{-}\mu Tp$:

$$\tau =_\mu \sigma \;:\Longleftrightarrow\; \mathbf{Tree}(\tau) = \mathbf{Tree}(\sigma)\,.$$

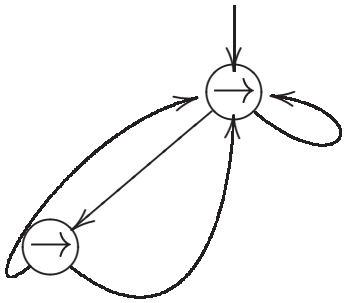
⊠

Example 3 (Tree Unfolding, Strong Recursive Type Equivalence).

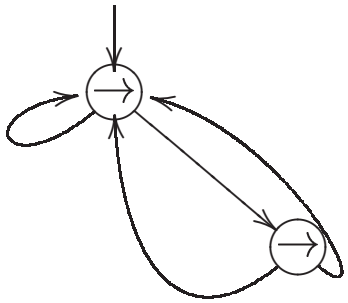
The recursive types

$$\tau := \mu\alpha.\,((\alpha \to \alpha) \to \alpha) \qquad \text{and} \qquad \sigma := \mu\alpha.\,(\alpha \to (\alpha \to \alpha))$$

correspond respectively to the different cyclic term graphs



and

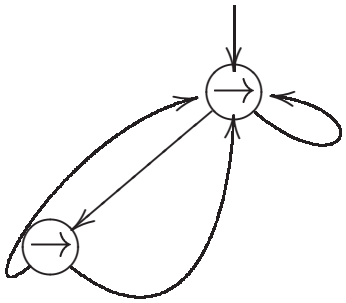


Example 3 (Tree Unfolding, Strong Recursive Type Equivalence).

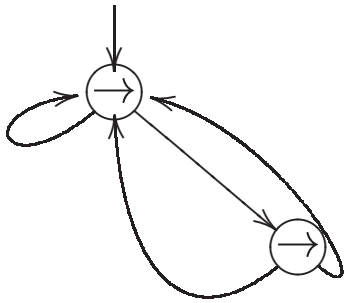
The recursive types

$$\tau := \mu\alpha.\,((\alpha \to \alpha) \to \alpha) \qquad \text{and} \qquad \sigma := \mu\alpha.\,(\alpha \to (\alpha \to \alpha))$$

correspond respectively to the different cyclic term graphs

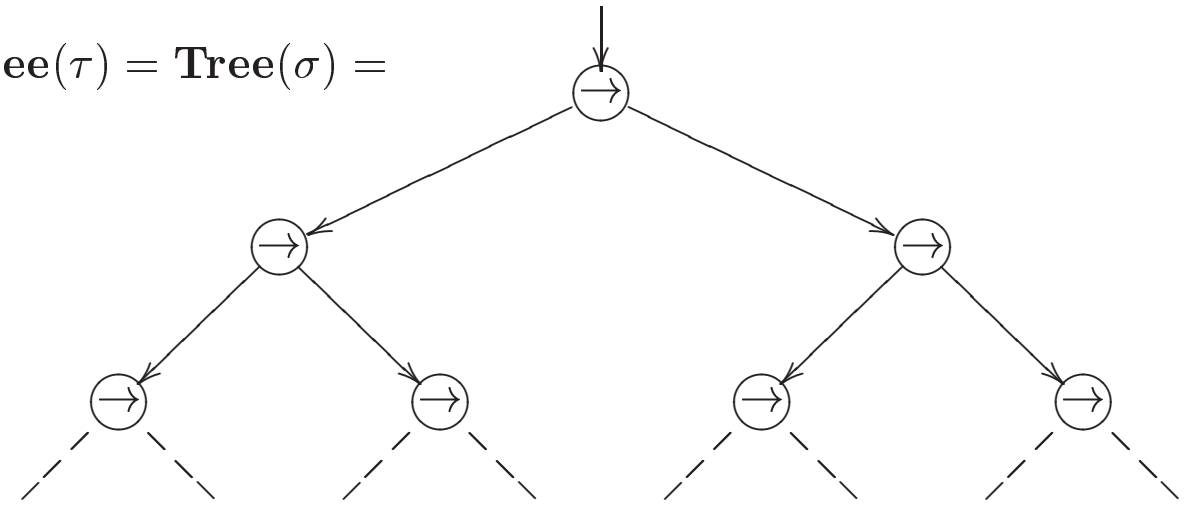


and



but possess the same tree unfolding

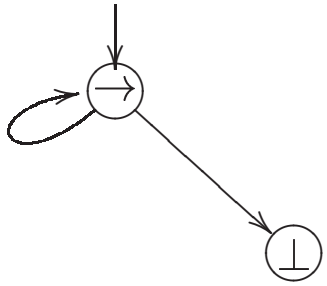
$\mathbf{Tree}(\tau) = \mathbf{Tree}(\sigma) =$



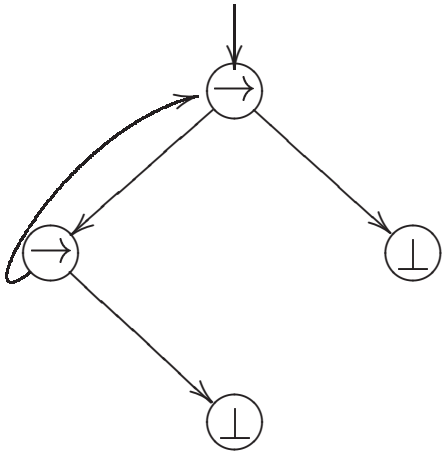
and hence $\tau =_\mu \sigma$ holds, i.e. recursive type equality holds between τ and σ. ■

Example 4 (Tree Unfolding, Strong Recursive Type Equivalence).

The recursive types $\tau := \mu\alpha.\,(\alpha \to \bot)$ and $\sigma := \mu\beta.\,((\beta \to \bot) \to \bot)$ correspond respectively to the different cyclic term graphs

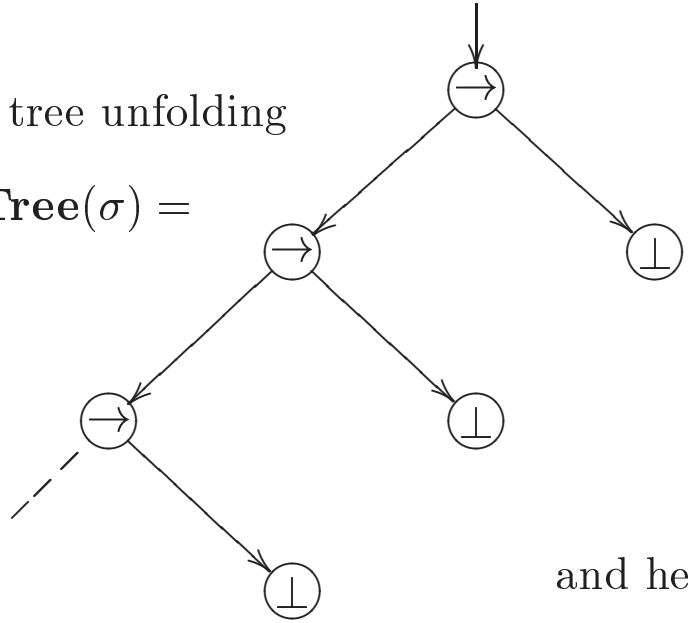


and



but possess the same tree unfolding

$\mathbf{Tree}(\tau) = \mathbf{Tree}(\sigma) =$



and hence it holds that $\tau =_\mu \sigma$. ■

Definition 5. (The variant system $\mathbf{HB}_{\mathbf{0}}^{=}$ of the axiomatization for $=_\mu$ given by Brandt and Henglein — 'Normalized type equality rules').

Axioms/Marked Assumptions:

$$\text{(REFL)}\quad \overline{\tau = \tau} \qquad \text{(Assm)}\quad (\tau = \sigma)^x \quad (\text{with } x \in Mk)\ .$$

Rules:

$$\frac{\tau = \sigma}{\tau' = \sigma'}\ \text{VAR}\quad (\text{where } \tau' \equiv_{\text{v}} \tau \text{ and } \sigma' \equiv_{\text{v}} \sigma)$$

$$\frac{\tau_0[\mu\alpha.\,\tau_0/\alpha] = \sigma}{\mu\alpha.\,\tau_0 = \sigma}\ \text{FOLD}_l \qquad \frac{\tau = \sigma_0[\mu\beta.\,\sigma_0/\beta]}{\tau = \mu\beta.\,\sigma_0}\ \text{FOLD}_r$$

$$\frac{\tau_1 = \sigma_1 \qquad\qquad \tau_2 = \sigma_2}{\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2}\ \text{ARROW}$$

⊠

Definition 5. (The variant system $\mathbf{HB}_{\mathbf{0}}^{=}$ of the axiomatization for $=_\mu$ given by Brandt and Henglein — 'Normalized type equality rules').

Axioms/Marked Assumptions:

$$\text{(REFL)}\quad \frac{}{\tau = \tau} \qquad \text{(Assm)}\quad (\tau = \sigma)^x \quad (\text{with } x \in Mk)\ .$$

Rules:

$$\frac{\tau = \sigma}{\tau' = \sigma'}\ \text{VAR} \quad (\text{where } \tau' \equiv_{\text{v}} \tau \text{ and } \sigma' \equiv_{\text{v}} \sigma)$$

$$\frac{\tau_0[\mu\alpha.\,\tau_0/\alpha] = \sigma}{\mu\alpha.\,\tau_0 = \sigma}\ \text{FOLD}_l \qquad \frac{\tau = \sigma_0[\mu\beta.\,\sigma_0/\beta]}{\tau = \mu\beta.\,\sigma_0}\ \text{FOLD}_r$$

$$\frac{\begin{array}{cc}\langle\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2\rangle^x & \langle\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2\rangle^x \\ \mathcal{D}_1 & \mathcal{D}_2 \\ \tau_1 = \sigma_1 & \tau_2 = \sigma_2\end{array}}{\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2}\ (\text{ARROW}/\text{FIX})_x \quad (\text{if side-cond. } \mathbf{I})$$

⊠

Definition 5. (The variant system $\mathbf{HB}_{\mathbf{0}}^{=}$ of the axiomatization for $=_\mu$ given by Brandt and Henglein — 'Normalized type equality rules').

Axioms/Marked Assumptions:

$$(\text{REFL}) \quad \frac{}{\tau = \tau} \qquad (\text{Assm}) \quad (\tau = \sigma)^x \quad (\text{with } x \in Mk) \ .$$

Rules:

$$\frac{\tau = \sigma}{\tau' = \sigma'} \text{ VAR} \quad (\text{where } \tau' \equiv_{\text{v}} \tau \text{ and } \sigma' \equiv_{\text{v}} \sigma)$$

$$\frac{\tau_0[\mu\alpha.\,\tau_0/\alpha] = \sigma}{\mu\alpha.\,\tau_0 = \sigma} \text{ FOLD}_l \qquad \frac{\tau = \sigma_0[\mu\beta.\,\sigma_0/\beta]}{\tau = \mu\beta.\,\sigma_0} \text{ FOLD}_r$$

$$\frac{\begin{array}{cc} \langle \tau_1 \to \tau_2 = \sigma_1 \to \sigma_2 \rangle^x & \langle \tau_1 \to \tau_2 = \sigma_1 \to \sigma_2 \rangle^x \\ \mathcal{D}_1 & \mathcal{D}_2 \\ \tau_1 = \sigma_1 & \tau_2 = \sigma_2 \end{array}}{\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2} \ (\text{ARROW}/\text{FIX})_x \ (\text{if side-cond. } \mathbf{I})$$

⊠

Theorem 6 ((∼[BrHe98]) Soundness and Completeness of $\mathbf{HB}_{\mathbf{0}}^{=}$ w.r.t. $=_\mu$).
For all recursive types $\tau, \sigma \in$ can-μTp it holds that:

$$\vdash_{\mathbf{HB}_{\mathbf{0}}^{=}} \tau = \sigma \quad \Longleftrightarrow \quad \tau =_\mu \sigma \ .$$

Definition 7. (The proof system $\mathbf{AK}_{\mathbf{0}}^{\equiv}$ à la Ariola, Klop — — 'Normalized syntactic-matching system').

Axioms: $\mathbf{AK}_{\mathbf{0}}^{\equiv}$ possesses **no** *axioms*.

Rules:

$$\frac{\mu\alpha.\,\tau_0 = \sigma}{\tau_0[\mu\alpha.\,\tau_0/\alpha] = \sigma}\ \text{UNFOLD}_l \qquad \frac{\tau = \mu\beta.\,\sigma_0}{\tau = \sigma_0[\mu\beta.\,\sigma_0/\beta]}\ \text{UNFOLD}_r$$

$$\frac{\tau = \sigma}{\tau' = \sigma'}\ \text{VAR} \quad (\text{if } \tau' \equiv_{\text{v}} \tau \text{ and } \sigma' \equiv_{\text{v}} \sigma) \qquad \frac{\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2}{\tau_i = \sigma_i}\ \text{DECOMP}\ \ (i = 1, 2)$$

⊠

Definition 7. (The proof system $\mathbf{AK}_{\mathbf{0}}^{=}$ à la Ariola, Klop — — 'Normalized syntactic-matching system').

Axioms: $\mathbf{AK}_{\mathbf{0}}^{=}$ possesses **no** *axioms*.

Rules:

$$\frac{\mu\alpha.\,\tau_0 = \sigma}{\tau_0[\mu\alpha.\,\tau_0/\alpha] = \sigma}\ \text{UNFOLD}_l \qquad \frac{\tau = \mu\beta.\,\sigma_0}{\tau = \sigma_0[\mu\beta.\,\sigma_0/\beta]}\ \text{UNFOLD}_r$$

$$\frac{\tau = \sigma}{\tau' = \sigma'}\ \text{VAR} \quad (\text{if } \tau' \equiv_{\text{v}} \tau \text{ and } \sigma' \equiv_{\text{v}} \sigma) \qquad \frac{\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2}{\tau_i = \sigma_i}\ \text{DECOMP}\ \ (i = 1, 2)$$

⊠

Theorem 9 ((∼[ArKl95]) Soundness and Completeness of $\mathbf{AK}_{\mathbf{0}}^{=}$ w.r.t. $=_\mu$).
For all recursive types $\tau, \sigma \in$ can-μTp:

$$\tau = \sigma \text{ is } \mathbf{AK}_{\mathbf{0}}^{=}\text{-consistent} \iff \tau =_\mu \sigma\ .$$

Definition 7. (The proof system $\mathbf{AK}_0^{=}$ à la Ariola, Klop — — 'Normalized syntactic-matching system').

Axioms: $\mathbf{AK}_0^{=}$ possesses **no** *axioms*.

Rules:

$$\frac{\mu\alpha.\,\tau_0 = \sigma}{\tau_0[\mu\alpha.\,\tau_0/\alpha] = \sigma}\ \text{UNFOLD}_l \qquad \frac{\tau = \mu\beta.\,\sigma_0}{\tau = \sigma_0[\mu\beta.\,\sigma_0/\beta]}\ \text{UNFOLD}_r$$

$$\frac{\tau = \sigma}{\tau' = \sigma'}\ \text{VAR} \quad (\text{if } \tau' \equiv_{\text{v}} \tau \text{ and } \sigma' \equiv_{\text{v}} \sigma) \qquad \frac{\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2}{\tau_i = \sigma_i}\ \text{DECOMP}\ \ (i = 1, 2)$$

⊠

Definition 8. For all recursive types $\tau, \sigma, \chi_1, \chi_2 \in \textit{can-}\mu Tp$:

(i) $\chi_1 = \chi_2$ is a *contradiction w.r.t.* $=_\mu$ $:\Longleftrightarrow$ $\mathcal{L}(\chi_1) \neq \mathcal{L}(\chi_2)$.

(ii) $\tau = \sigma$ is $\mathbf{AK}_0^{=}$*-consistent* $:\Longleftrightarrow$ no contradiction w.r.t. $=_\mu$ is derivable in $\mathbf{AK}_0^{=}$ from the assumption $\tau = \sigma$.

⊠

Theorem 9 ((∼[ArKl95]) Soundness and Completeness of $\mathbf{AK}_0^{=}$ w.r.t. $=_\mu$).
For all recursive types $\tau, \sigma \in \textit{can-}\mu Tp$:

$$\tau = \sigma \textit{ is } \mathbf{AK}_0^{=}\textit{-consistent} \iff \tau =_\mu \sigma \ .$$

The basic observation [due to J.W. Klop].

Example 10 ('Unfolding' consistency with respect to $\mathbf{AK}_{\mathbf{0}}^{=}$ by 'Loop Checking'). We consider again the recursive types in canonical form

$$\tau := \mu\alpha.\,((\alpha \to \alpha) \to \alpha) \qquad \text{and} \qquad \sigma := \mu\alpha.\,(\alpha \to (\alpha \to \alpha))$$

and try to establish the consistency of the equality $\tau = \sigma$ with respect to $\mathbf{AK}_{\mathbf{0}}^{=}$: For this we assemble all possible initial sequences of derivations in $\mathbf{AK}_{\mathbf{0}}^{=}$ (without applications of VAR) until *looping* occurs:

■

The basic observation [due to J.W. Klop].

Example 10 ('Unfolding' consistency with respect to $\mathbf{AK}_0^{=}$ by 'Loop Checking'). We consider again the recursive types in canonical form

$$\tau := \mu\alpha.\,((\alpha \to \alpha) \to \alpha) \qquad \text{and} \qquad \sigma := \mu\alpha.\,(\alpha \to (\alpha \to \alpha))$$

and try to establish the consistency of the equality $\tau = \sigma$ with respect to $\mathbf{AK}_0^{=}$: For this we assemble all possible initial sequences of derivations in $\mathbf{AK}_0^{=}$ (without applications of VAR) until *looping* occurs:

$$\dfrac{\Big(\overbrace{\mu\alpha.\,((\alpha \to \alpha) \to \alpha)}^{\equiv:\,\tau} = \overbrace{\mu\alpha.\,(\alpha \to (\alpha \to \alpha))}^{\equiv:\,\sigma}\Big)^{y}}{(\tau \to \tau) \to \tau = \sigma \to (\sigma \to \sigma)}\ \text{UNFOLD}_{l/r}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{(\tau \to \tau = \sigma)^{x}}{\tau \to \tau = \sigma \to (\sigma \to \sigma)}}{(\tau = \sigma)^{y}} \qquad \dfrac{\dfrac{\tau = \sigma \to \sigma}{(\tau \to \tau) \to \tau = \sigma \to \sigma}}{(\tau \to \tau = \sigma)^{x} \qquad (\tau = \sigma)^{y}}}{(\tau \to \tau) \to \tau = \sigma \to (\sigma \to \sigma)}\ \text{DECOMP} \qquad \dfrac{\dfrac{\dfrac{\dfrac{(\tau = \sigma \to \sigma)^{z}}{(\tau \to \tau) \to \tau = \sigma \to \sigma}}{\tau \to \tau = \sigma}}{\tau \to \tau = \sigma \to (\sigma \to \sigma)}}{(\tau = \sigma)^{y} \qquad (\tau = \sigma \to \sigma)^{z}}\ (\tau = \sigma)^{y}$$

■

The basic observation (continued).

Plain Reflection in upwards-direction gives:

$$\dfrac{\dfrac{\dfrac{(\tau \to \tau = \sigma)^x \quad (\tau = \sigma)^y}{(\tau \to \tau) \to \tau = \sigma \to \sigma} \quad \dfrac{(\tau = \sigma)^y \quad (\tau = \sigma \to \sigma)^z}{\tau \to \tau = \sigma \to (\sigma \to \sigma)}}{\dfrac{(\tau = \sigma)^y \qquad \tau = \sigma \to \sigma}{\tau \to \tau = \sigma \to (\sigma \to \sigma)}} \quad \dfrac{\tau \to \tau = \sigma \qquad (\tau = \sigma)^y}{\dfrac{(\tau \to \tau) \to \tau = \sigma \to \sigma}{(\tau = \sigma \to \sigma)^z}}}{\dfrac{(\tau \to \tau = \sigma)^x}{(\tau \to \tau) \to \tau = \sigma \to (\sigma \to \sigma)}}\,\text{ARROW}}{\Big(\underbrace{\mu\alpha.\,((\alpha \to \alpha) \to \alpha)}_{\equiv:\,\tau} = \underbrace{\mu\alpha.\,(\alpha \to (\alpha \to \alpha))}_{\equiv:\,\sigma}\Big)^y}\,\text{FOLD}_{l/r}$$

The basic observation (continued).

Example 11 (A Proof in $\mathbf{HB}_{\mathbf{0}}^{=}$).
The following is a derivation in $\mathbf{HB}_{\mathbf{0}}^{=}$ with conclusion $\tau = \sigma$ (with τ and σ as before) and without open assumption classes:

$$\dfrac{\dfrac{\dfrac{\dfrac{(\dots)^{y}}{\tau = \sigma} \quad \dfrac{\dfrac{(\dots)^{x}}{\tau \to \tau = \sigma} \quad \dfrac{(\dots)^{y}}{\tau = \sigma}}{(\tau \to \tau) \to \tau = \sigma \to \sigma}}{\tau = \sigma \to \sigma}\,\text{X}}{\tau \to \tau = \sigma \to (\sigma \to \sigma)} \quad \dfrac{\dfrac{\dfrac{(\dots)^{y}}{\tau = \sigma} \quad \dfrac{(\dots)^{z}}{\tau = \sigma \to \sigma}}{\tau \to \tau = \sigma \to (\sigma \to \sigma)} \quad \dfrac{(\dots)^{y}}{\tau = \sigma}}{(\tau \to \tau) \to \tau = \sigma \to \sigma}\,\text{Z}}{\tau \to \tau = \sigma}}{\tau = \sigma \to \sigma}\,(\text{ARR./FIX})_{y}}{(\tau \to \tau) \to \tau = \sigma \to (\sigma \to \sigma)}\,\text{FOLD}_{l/r}}{\underbrace{\mu\alpha.\,((\alpha \to \alpha) \to \alpha)}_{\equiv:\,\tau} = \underbrace{\mu\alpha.\,(\alpha \to (\alpha \to \alpha))}_{\equiv:\,\sigma}}$$

■

The Formalization: Step 1.

Definition 12. (The extension e-HB$_0^=$ of the system HB$_0^=$). The extension **e-HB$_0^=$** of the system **HB$_0^=$** possesses the same formulas and axioms as **HB$_0^=$**; it contains all derivation rules of **HB$_0^=$**, but additionally also the following three:

$$\frac{\begin{array}{c}[\tau' = \sigma']^x \\ \mathcal{D}_0 \\ \tau = \sigma\end{array}}{\tau' = \sigma'}\ (\text{VAR/FIX})_x \qquad \begin{array}{l}(\text{where } \tau' \equiv_{\text{v}} \tau \text{ and } \sigma' \equiv_{\text{v}} \sigma \\ \text{and if side-conditions } \mathbf{C} \text{ and } \mathbf{I})\end{array}$$

Hereby:

- *Side-condition* **C**: $\mathcal{D}_0$ is *contractive* w.r.t. the marked open assumptions $(\dots)^x$, which get discharged in the respective rule application.
- *Side-condition* **I**: The discharged class of assms. $[\tau' = \sigma']^x$ in $\mathcal{D}_0$ is *inhabited*.

⊠

The Formalization: Step 1.

Definition 12. (The extension e-HB$_0^=$ of the system HB$_0^=$). The extension **e-HB$_0^=$** of the system **HB$_0^=$** possesses the same formulas and axioms as **HB$_0^=$**; it contains all derivation rules of **HB$_0^=$**, but additionally also the following three:

$$\frac{\begin{array}{c}[\tau' = \sigma']^x \\ \mathcal{D}_0 \\ \tau = \sigma\end{array}}{\tau' = \sigma'}\ (\text{VAR/FIX})_x \qquad \begin{array}{l}(\text{where } \tau' \equiv_{\text{v}} \tau \text{ and } \sigma' \equiv_{\text{v}} \sigma \\ \text{and if side-conditions } \mathbf{C} \text{ and } \mathbf{I})\end{array}$$

$$\frac{\begin{array}{c}\big[\mu\alpha.\,\tau_0 = \sigma_0\big]^x \\ \mathcal{D}_0 \\ \tau_0[\mu\alpha.\,\tau_0/\alpha] = \sigma_0\end{array}}{\mu\alpha.\,\tau_0 = \sigma_0}\ (\text{FOLD}_l/\text{FIX})_x \qquad (\text{if side-condition } \mathbf{I})$$

and FOLD$_r$/FIX, which is of an analogous form as FOLD$_l$/FIX. Hereby:

- *Side-condition* **C**: $\mathcal{D}_0$ is *contractive* w.r.t. the marked open assumptions $(\dots)^x$, which get discharged in the respective rule application.
- *Side-condition* **I**: The discharged class of assms. $[\tau' = \sigma']^x$ in $\mathcal{D}_0$ is *inhabited*.

⊠

Lemma 13. (Equivalence of the systems $\mathbf{HB}_{\mathbf{0}}^{=}$ and e-$\mathbf{HB}_{\mathbf{0}}^{=}$).

(i)

(ii)

(iii) *As a consequence, the systems* **e-HB**$_{\mathbf{0}}^{=}$ *and* $\mathbf{HB}_{\mathbf{0}}^{=}$ *are equivalent, i.e. they possess the same theorems. Thus* **e-HB**$_{\mathbf{0}}^{=}$ *is a conservative extension of* $\mathbf{HB}_{\mathbf{0}}^{=}$.

Lemma 13. (Equivalence of the systems $\mathbf{HB}_{\mathbf{0}}^{=}$ and e-$\mathbf{HB}_{\mathbf{0}}^{=}$).

(i) *The rules VAR/FIX, $FOLD_l$/FIX and $FOLD_r$/FIX of* **e-$\mathbf{HB}_{\mathbf{0}}^{=}$** *are admissible rules of the system* **$\mathbf{HB}_{\mathbf{0}}^{=}$**.

(ii)

(iii) *As a consequence, the systems* **e-$\mathbf{HB}_{\mathbf{0}}^{=}$** *and* **$\mathbf{HB}_{\mathbf{0}}^{=}$** *are equivalent, i.e. they possess the same theorems. Thus* **e-$\mathbf{HB}_{\mathbf{0}}^{=}$** *is a conservative extension of* **$\mathbf{HB}_{\mathbf{0}}^{=}$**.

Lemma 13. (Equivalence of the systems $\mathbf{HB}_{\mathbf{0}}^{=}$ and e-$\mathbf{HB}_{\mathbf{0}}^{=}$).

(i) The rules VAR/FIX, $FOLD_l$/FIX and $FOLD_r$/FIX of e-$\mathbf{HB}_{\mathbf{0}}^{=}$ *are admissible rules of the system* $\mathbf{HB}_{\mathbf{0}}^{=}$.

(ii) What is more: Every derivation $\mathcal{D}$ in e-$\mathbf{HB}_{\mathbf{0}}^{=}$ *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.*

(iii) As a consequence, the systems e-$\mathbf{HB}_{\mathbf{0}}^{=}$ *and* $\mathbf{HB}_{\mathbf{0}}^{=}$ *are equivalent, i.e. they possess the same theorems. Thus* e-$\mathbf{HB}_{\mathbf{0}}^{=}$ *is a conservative extension of* $\mathbf{HB}_{\mathbf{0}}^{=}$.

The Formalization: Step 2.

Definition 14. (Partial Consistency-Unfoldings in $\mathbf{AK}_{\mathbf{0}}^{=}$). For all $\tau, \sigma \in \mathit{can}\text{-}\mu Tp$ a *partial consistency-unfolding* (a *p.c.u.*) $\mathcal{C}$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ of the equation $\tau = \sigma$ is a 'derivation tree' that can be formed together with the assertion "$\mathcal{C}$ is a p.c.u. of $\tau = \sigma$ in $\mathbf{AK}_{\mathbf{0}}^{=}$" according to the generation rules (i)–(v) below: The notion of a *'unbound leaf-occurrence of a marked formula'* (an *u.l.o.m.f.*) in a p.c.u. is defined in parallel:

(i) For all $\tau, \sigma \in \mathit{can}\text{-}\mu Tp$ $\boxed{(\tau = \sigma)^x}$ is a p.c.u. $\mathcal{C}$ from $\tau = \sigma$. The occurrence of $(\tau = \sigma)^x$ in $\mathcal{C}$ is the single u.l.o.m.f. in $\mathcal{C}$. – Furthermore for all $\tau \in \mathit{can}\text{-}\mu Tp$ $\boxed{\tau = \tau}$ is a p.c.u. of $\tau = \tau$, which contains no u.l.o.m.f.'s in $\mathcal{C}$.

The Formalization: Step 2.

Definition 14. (Partial Consistency-Unfoldings in $\mathbf{AK}_{\mathbf{0}}^{=}$). For all $\tau, \sigma \in$ *can-*μTp a *partial consistency-unfolding* (a *p.c.u.*) $\mathcal{C}$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ of the equation $\tau = \sigma$ is a 'derivation tree' that can be formed together with the assertion "$\mathcal{C}$ is a p.c.u. of $\tau = \sigma$ in $\mathbf{AK}_{\mathbf{0}}^{=}$" according to the generation rules (i)–(v) below: The notion of a *'unbound leaf-occurrence of a marked formula'* (an *u.l.o.m.f.*) in a p.c.u. is defined in parallel:

(i) For all $\tau, \sigma \in$ *can-*μTp $\boxed{(\tau = \sigma)^x}$ is a p.c.u. $\mathcal{C}$ from $\tau = \sigma$. The occurrence of $(\tau = \sigma)^x$ in $\mathcal{C}$ is the single u.l.o.m.f. in $\mathcal{C}$. – Furthermore for all $\tau \in$ *can-*μTp $\boxed{\tau = \tau}$ is a p.c.u. of $\tau = \tau$, which contains no u.l.o.m.f.'s in $\mathcal{C}$.

(ii) For all $\tau, \sigma, \tau_0, \sigma_0 \in$ *can-*μTp $\boxed{\dfrac{\dfrac{\tau = \sigma}{(\tau_0 = \sigma_0)^{\boldsymbol{m}_0}}\, R}{\mathcal{C}_0}}$ is a p.c.u. $\mathcal{C}$ of $\tau = \sigma$, given that $\mathcal{C}_0$ is a p.c.u. of $\tau_0 = \sigma_0$ and that R is an application of a rule $\mathrm{UNFOLD}_{l/r}$ or VAR. An u.l.o.m.f. in $\mathcal{C}$ is such an occurrence of a marked formula in $\mathcal{C}$ within its subtree $\mathcal{C}_0$ that corresponds to an u.l.o.m.f. in $\mathcal{C}_0$.

(iii) (a) For all $\tau_0, \sigma \in \mathit{can}\text{-}\mu Tp$ and $\alpha \in TVar$

$$\frac{(\mu\alpha.\,\tau_0 = \sigma)^x}{(\tau_0[\mu\alpha.\,\tau_0/\alpha] = \sigma)^{m_0}}\ \text{UNFOLD}_l$$

$$\mathcal{C}_0$$

$$[\mu\alpha.\,\tau_0 = \sigma]^x$$

is a p.c.u. $\mathcal{C}$ of $\mu\alpha.\,\tau_0 = \sigma$, given that $\mathcal{C}_0$ is a p.c.u. of $\tau_0[\mu\alpha.\,\tau_0/\alpha] = \sigma$ in which the (indicated) class $[\tau_0[\mu\alpha.\,\tau_0/\alpha] = \sigma]^x$ of all u.l.o.m.f.'s of the form $(\tau_0[\mu\alpha.\,\tau_0/\alpha] = \sigma)^x$ is non-empty.

(iii) (a) For all $\tau_0, \sigma \in \mathit{can}\text{-}\mu Tp$ and $\alpha \in TVar$

$$\frac{(\mu\alpha.\,\tau_0 = \sigma)^x}{\begin{array}{c}(\tau_0[\mu\alpha.\,\tau_0/\alpha] = \sigma)^{m_0} \\ \mathcal{C}_0 \\ [\mu\alpha.\,\tau_0 = \sigma]^x\end{array}}\ \mathrm{UNFOLD}_l$$

is a p.c.u. $\mathcal{C}$ of $\mu\alpha.\,\tau_0 = \sigma$, given that $\mathcal{C}_0$ is a p.c.u. of $\tau_0[\mu\alpha.\,\tau_0/\alpha] = \sigma$ in which the (indicated) class $[\tau_0[\mu\alpha.\,\tau_0/\alpha] = \sigma]^x$ of all u.l.o.m.f.'s of the form $(\tau_0[\mu\alpha.\,\tau_0/\alpha] = \sigma)^x$ is non-empty.

All occurrences of $(\mu\alpha.\,\tau_0 = \sigma)^x$ within the subtree $\mathcal{C}_0$ of $\mathcal{C}$, that correspond to u.l.o.m.f.'s in $\mathcal{C}_0$, are *bound back* in $\mathcal{C}$ to the occurrence of $(\mu\alpha.\,\tau_0 = \sigma)^x$ at the root. For all marked formulas $(\tilde{\tau} = \tilde{\sigma})^{\tilde{x}}$ different from $(\mu\alpha.\,\tau_0 = \sigma)^x$ the unbound leaf-occurrences of this marked formula correspond uniquely to the u.l.o.'s of $(\tilde{\tau} = \tilde{\sigma})^{\tilde{x}}$ within the subtree $\mathcal{C}_0$ of $\mathcal{C}$.

(iii) (a) For all $\tau_0, \sigma \in \mathit{can}\text{-}\mu Tp$ and $\alpha \in TVar$

$$\frac{(\mu\alpha.\,\tau_0 = \sigma)^x}{\begin{array}{c}(\tau_0[\mu\alpha.\,\tau_0/\alpha] = \sigma)^{m_0} \\ \mathcal{C}_0 \\ [\mu\alpha.\,\tau_0 = \sigma]^x\end{array}}\ \mathrm{UNFOLD}_l$$

is a p.c.u. $\mathcal{C}$ of $\mu\alpha.\,\tau_0 = \sigma$, given that $\mathcal{C}_0$ is a p.c.u. of $\tau_0[\mu\alpha.\,\tau_0/\alpha] = \sigma$ in which the (indicated) class $[\tau_0[\mu\alpha.\,\tau_0/\alpha] = \sigma]^x$ of all u.l.o.m.f.'s of the form $(\tau_0[\mu\alpha.\,\tau_0/\alpha] = \sigma)^x$ is non-empty.
All occurrences of $(\mu\alpha.\,\tau_0 = \sigma)^x$ within the subtree $\mathcal{C}_0$ of $\mathcal{C}$, that correspond to u.l.o.m.f.'s in $\mathcal{C}_0$, are *bound back* in $\mathcal{C}$ to the occurrence of $(\mu\alpha.\,\tau_0 = \sigma)^x$ at the root. For all marked formulas $(\tilde{\tau} = \tilde{\sigma})^{\tilde{x}}$ different from $(\mu\alpha.\,\tau_0 = \sigma)^x$ the unbound leaf-occurrences of this marked formula correspond uniquely to the u.l.o.'s of $(\tilde{\tau} = \tilde{\sigma})^{\tilde{x}}$ within the subtree $\mathcal{C}_0$ of $\mathcal{C}$.

(b) An analogous clause with an application of UNFOLD_r at the top.

(c) A similar clause with an application of VAR at the top: Here a small *formal complication* comes in and has to be treated (*'contractivity'* of $\mathcal{C}_0$).

(iv)
$$\frac{\tau_{01} \to \tau_{02} = \sigma_{01} \to \sigma_{02}}{\underset{\mathcal{C}_{01}}{(\tau_{01} = \sigma_{01})^{m_{01}}} \qquad \underset{\mathcal{C}_{02}}{(\tau_{02} = \sigma_{02})^{m_{02}}}}\ \text{DECOMP}$$
is a p.c.u. $\mathcal{C}$ of

$\tau_{01} \to \tau_{02} = \sigma_{01} \to \sigma_{02}$ given that $\mathcal{C}_{0i}$ is a p.c.u. of $\tau_{0i} = \sigma_{0i}$ for each $i \in \{1, 2\}$. The u.l.o.m.f.'s in $\mathcal{C}$ correspond uniquely to the u.l.o.m.f.'s in either of its immediate subtrees $\mathcal{C}_{01}$ or $\mathcal{C}_{02}$.

(v)
$$\frac{(\tau_{01} \to \tau_{02} = \sigma_{01} \to \sigma_{02})^{x}}{\underset{\substack{\mathcal{C}_{01} \\ \langle \tau = \sigma \rangle^{x}}}{(\tau_{01} = \sigma_{01})^{m_{01}}} \qquad \underset{\substack{\mathcal{C}_{02} \\ \langle \tau = \sigma \rangle^{x}}}{(\tau_{02} = \sigma_{02})^{m_{02}}}}\ \text{DECOMP}$$
is a p.c.u. $\mathcal{C}$ of

$\tau_{01} \to \tau_{02} = \sigma_{01} \to \sigma_{02}$, given that $\mathcal{C}_{0i}$ is a p.c.u. of $\tau_{0i} = \sigma_{0i}$ for each $i \in \{1, 2\}$ and that there is at least one unbound leaf-occurrence of the marked formula $(\tau_{01} \to \tau_{02} = \sigma_{01} \to \sigma_{02})^{x}$ in either $\mathcal{C}_{01}$ or in $\mathcal{C}_{02}$.

[... similar clause about the u.l.o.m.f.'s as in (iii) ...].

The *depth* $|\mathcal{C}|$ of a p.c.u. $\mathcal{C}$ is defined as the depth of the underlying (derivation-) tree. ⊠

The Formalization: Step 2 (continued).

Definition 15. (Consistency-Unfoldings in $\mathbf{AK_0^=}$). Let $\tau, \sigma \in \mathit{can}\text{-}\mu Tp$. A partial consistency-unfolding $\mathcal{C}$ of $\tau = \sigma$ in $\mathbf{AK_0^=}$ is called a *consistency-unfolding* (a *c.u.*) of $\tau = \sigma$ in $\mathbf{AK_0^=}$ if and only if $\mathcal{C}$ does not contain any unbound leaf-occurrences of marked formulas.

The Formalization: Step 2 (continued).

Definition 15. (Consistency-Unfoldings in $\mathbf{AK_0^=}$). Let $\tau, \sigma \in \mathit{can}\text{-}\mu Tp$. A partial consistency-unfolding $\mathcal{C}$ of $\tau = \sigma$ in $\mathbf{AK_0^=}$ is called a *consistency-unfolding* (a *c.u.*) of $\tau = \sigma$ in $\mathbf{AK_0^=}$ if and only if $\mathcal{C}$ does not contain any unbound leaf-occurrences of marked formulas.

For example, the derivation tree encountered earlier in Example 10 can be recognized as a consistency-unfolding in $\mathbf{AK_0^=}$.

The Formalization: Step 2 (continued).

Definition 15. (Consistency-Unfoldings in $\mathbf{AK}_{\mathbf{0}}^{=}$). Let $\tau, \sigma \in \mathit{can}\text{-}\mu Tp$. A partial consistency-unfolding $\mathcal{C}$ of $\tau = \sigma$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ is called a *consistency-unfolding* (a *c.u.*) of $\tau = \sigma$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ if and only if $\mathcal{C}$ does not contain any unbound leaf-occurrences of marked formulas.

For example, the derivation tree encountered earlier in Example 10 can be recognized as a consistency-unfolding in $\mathbf{AK}_{\mathbf{0}}^{=}$.

Theorem 16. *For all recursive types $\tau, \sigma \in \mathit{can}\text{-}\mu Tp$ it holds that:*

$$\tau = \sigma \textit{ is } \mathbf{AK}_{\mathbf{0}}^{=}\textit{-consistent} \iff \textit{There exists a consistency-unfolding of } \tau = \sigma \textit{ in } \mathbf{AK}_{\mathbf{0}}^{=} .$$

The Formalization: Step 3.

Definition 17. (The Reflection Mappings $\mathcal{D}(\cdot)$ and $\mathcal{C}(\cdot)$). Three exemplary cases of inductive clauses in the definitions of the reflection mappings $\mathcal{D}(\cdot)$ and $\mathcal{C}(\cdot)$ between partial consistency-unfoldings in $\mathbf{AK}_{\mathbf{0}}^{=}$ and derivations in $\mathbf{e\text{-}HB}_{\mathbf{0}}^{=}$:

$$\boxed{(\tau = \sigma)^{\boldsymbol{m}}} \quad \overset{\mathcal{D}(\cdot)}{\underset{\mathcal{C}(\cdot)}{\longmapsto\!\!\!\!\longleftarrow}} \quad \boxed{(\tau = \sigma)^{\boldsymbol{m}}}$$

$$\boxed{\dfrac{\begin{array}{c}(\mu\alpha.\,\tau_0 = \sigma)^{x} \\ (\tau_0[\mu\alpha.\,\tau_0/\alpha] = \sigma)^{\boldsymbol{m}_0} \\ \tilde{\mathcal{C}}_0 \quad (\equiv \mathcal{C}(\tilde{\mathcal{D}}_0)) \\ [\mu\alpha.\,\tau_0 = \sigma]^{x}\end{array}}{}\ \mathrm{UNFOLD}_l} \quad \overset{\mathcal{D}(\cdot)}{\underset{\mathcal{C}(\cdot)}{\longmapsto\!\!\!\!\longleftarrow}} \quad \boxed{\dfrac{\begin{array}{c}[\mu\alpha.\,\tau_0 = \sigma]^{x} \\ \tilde{\mathcal{D}}_0 \quad (\equiv \mathcal{D}(\tilde{\mathcal{C}}_0)) \\ \tau_0[\mu\alpha.\,\tau_0/\alpha] = \sigma\end{array}}{\mu\alpha.\,\tau_0 = \sigma}\ (\mathrm{FOLD}_l/\mathrm{FIX})_{x}}$$

The Formalization: Step 3.

Definition 17. (The Reflection Mappings $\mathcal{D}(\cdot)$ and $\mathcal{C}(\cdot)$). Three exemplary cases of inductive clauses in the definitions of the reflection mappings $\mathcal{D}(\cdot)$ and $\mathcal{C}(\cdot)$ between partial consistency-unfoldings in $\mathbf{AK}_{\mathbf{0}}^{=}$ and derivations in $\mathbf{e\text{-}HB}_{\mathbf{0}}^{=}$:

$$\boxed{(\tau = \sigma)^{\boldsymbol{m}}} \quad \overset{\mathcal{D}(\cdot)}{\underset{\mathcal{C}(\cdot)}{\longmapsto\!\!\!\!\longleftarrow}} \quad \boxed{(\tau = \sigma)^{\boldsymbol{m}}}$$

$$\boxed{\dfrac{\begin{array}{c}(\mu\alpha.\,\tau_0 = \sigma)^{x} \\ (\tau_0[\mu\alpha.\,\tau_0/\alpha] = \sigma)^{\boldsymbol{m}_0}\end{array}\ \text{UNFOLD}_l \\ \begin{array}{c}\tilde{\mathcal{C}}_0 \quad (\equiv \mathcal{C}(\tilde{\mathcal{D}}_0)) \\ [\mu\alpha.\,\tau_0 = \sigma]^{x}\end{array}} \quad \overset{\mathcal{D}(\cdot)}{\underset{\mathcal{C}(\cdot)}{\longmapsto\!\!\!\!\longleftarrow}} \quad \boxed{\begin{array}{c}[\mu\alpha.\,\tau_0 = \sigma]^{x} \\ \tilde{\mathcal{D}}_0 \quad (\equiv \mathcal{D}(\tilde{\mathcal{C}}_0)) \\ \dfrac{\tau_0[\mu\alpha.\,\tau_0/\alpha] = \sigma}{\mu\alpha.\,\tau_0 = \sigma}\ (\text{FOLD}_l/\text{FIX})_{x}\end{array}}$$

$$\boxed{\dfrac{\begin{array}{c}\tau_{01} \to \tau_{02} = \sigma_{01} \to \sigma_{02} \\ (\tau_{01} = \sigma_{01})^{\boldsymbol{m}_{01}} \qquad\qquad (\tau_{02} = \sigma_{02})^{\boldsymbol{m}_{02}}\end{array}\ \text{DECOMP} \\ \begin{array}{cc}\tilde{\mathcal{C}}_{01} \quad (\equiv \mathcal{C}(\tilde{\mathcal{D}}_{01})) & \tilde{\mathcal{C}}_{02} \quad (\equiv \mathcal{C}(\tilde{\mathcal{D}}_{02}))\end{array}} \quad \overset{\mathcal{D}(\cdot)}{\underset{\mathcal{C}(\cdot)}{\longmapsto\!\!\!\!\longleftarrow}}$$

$$\longmapsto\!\!\!\!\longleftarrow \quad \boxed{\dfrac{\begin{array}{cc}\tilde{\mathcal{D}}_{01} \quad (\equiv \mathcal{D}(\tilde{\mathcal{C}}_{01})) & \tilde{\mathcal{D}}_{02} \quad (\equiv \mathcal{D}(\tilde{\mathcal{C}}_{02})) \\ \tau_{01} = \sigma_{01} & \tau_{02} = \sigma_{02}\end{array}}{\tau_{01} \to \tau_{02} = \sigma_{01} \to \sigma_{02}}\ \text{ARROW}} \qquad \boxtimes$$

The Formalization: Main Theorem.

Theorem 18. (A Duality between derivations in e-HB$_0^=$ and consistency-unfoldings in AK$_0^=$).

There is a bijective functional relationship *between derivations in* **e-HB$_0^=$** *and consistency-unfoldings in* **AK$_0^=$** *via the reflection mappings* $\mathcal{D}(\cdot)$ *and* $\mathcal{C}(\cdot)$ *of Definition 17 in the following sense:*

— *For all* $\tau, \sigma \in$ *can-*μTp*:*

$$\boxed{\begin{array}{c}\tilde{\mathcal{D}}\\ \tau = \sigma\end{array}} \equiv \mathcal{D}(\tilde{\mathcal{C}}) \qquad \begin{array}{c}\mathcal{C}(\cdot)\\ \longmapsto\\ \longleftarrow\\ \mathcal{D}(\cdot)\end{array} \qquad \mathcal{C}(\tilde{\mathcal{D}}) \equiv \boxed{\begin{array}{c}\tau = \sigma\\ \tilde{\mathcal{C}}\end{array}}$$

e-HB$_0^=$-*derivation of* $\tau = \sigma$ *without open assumption classes*

consistency-unfolding of $\tau = \sigma$ *in* **AK$_0^=$**

The Formalization: Main Theorem.

Theorem 18. (A Duality between derivations in e-HB$_0^=$ and consistency-unfoldings in AK$_0^=$).

There is a bijective functional relationship *between derivations in* **e-HB$_0^=$** *and consistency-unfoldings in* **AK$_0^=$** *via the reflection mappings* $\mathcal{D}(\cdot)$ *and* $\mathcal{C}(\cdot)$ *of Definition 17 in the following sense:*

— *For all* $\tau, \sigma \in$ *can-*μTp*:*

$$\boxed{\begin{array}{c}\tilde{\mathcal{D}}\\ \tau = \sigma\end{array}} \equiv \mathcal{D}(\tilde{\mathcal{C}}) \qquad \begin{array}{c}\mathcal{C}(\cdot)\\ \longmapsto\\ \longleftarrow\\ \mathcal{D}(\cdot)\end{array} \qquad \mathcal{C}(\tilde{\mathcal{D}}) \equiv \boxed{\begin{array}{c}\tau = \sigma\\ \tilde{\mathcal{C}}\end{array}}$$

e-HB$_0^=$-*derivation of* $\tau = \sigma$ *without open assumption classes*

consistency-unfolding of $\tau = \sigma$ *in* **AK$_0^=$**

— *Thereby the reflection functions* $\mathcal{C}(\cdot)$ *and* $\mathcal{D}(\cdot)$ *are each other's inverses.*

The very immediate kind of this bijective functional relationship and the possibility to visualize it geometrically is reason to call it a duality.

Proof. By induction on the depth $|\tilde{\mathcal{D}}|$ of arbitrary **e-HB$_\mathbf{0}^=$**-derivations $\tilde{\mathcal{D}}$ with n open assumption classes and respectively by induction on the depth $|\tilde{\mathcal{C}}|$ of arbitrary partial consistency-unfoldings $\tilde{\mathcal{C}}$ in **AK$_\mathbf{0}^=$** with n classes of unbound leaf-occurrences of marked formulas (where $n \in \mathbb{N}_0$):

$$\begin{array}{c} \{\, [\tau_i = \sigma_i]^{x_i} \,\}_{i=1,\ldots,n} \\ \tilde{\mathcal{D}} \\ \tau = \sigma \end{array}$$

e-HB$_\mathbf{0}^=$-derivation of $\tau = \sigma$ with n open assumption classes

$\overset{\mathcal{C}(\cdot)}{\longmapsto}$
$\overset{\mathcal{D}(\cdot)}{\longleftarrow}$

$$\begin{array}{c} (\tau = \sigma)^{\boldsymbol{m}} \\ \tilde{\mathcal{C}} \\ \{\, [\tau_i = \sigma_i]^{x_i} \,\}_{i=1,\ldots,n} \end{array}$$

partial consistency-unfolding of $\tau = \sigma$ in **AK$_\mathbf{0}^=$** with n classes of unbound leaf-occurrences of marked formulas

□

Example 19. (Duality between a derivation in e-HB$_0^{=}$ and a consistency-unfolding in AK$_0^{=}$).

$$\dfrac{\dfrac{\dfrac{\overbrace{\left(\mu\alpha.\,(\alpha \to \bot)\right)}^{\equiv:\,\tau} = \overbrace{\left(\mu\beta.\,((\beta \to \bot) \to \bot)\right)}^{\equiv:\,\sigma\;x}}{\tau \to \bot = \sigma}\,\text{UNFOLD}_l}{\tau \to \bot = (\sigma \to \bot) \to \bot}\,\text{UNFOLD}_r \qquad \dfrac{\tau = \sigma \to \bot}{\tau \to \bot = \sigma \to \bot}}{\left(\tau = \sigma\right)^x \qquad \bot = \bot}\,\text{DECOMP} \qquad \bot = \bot\,\text{DECOMP}$$

■

Example 19. (Duality between a derivation in e-HB$_0^{=}$ and a consistency-unfolding in AK$_0^{=}$).

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{(\tau = \sigma)^{x}}{\tau \to \bot = \sigma \to \bot}\,\text{ARROW}}{\tau = \sigma \to \bot} \qquad \dfrac{\bot = \bot}{\bot = \bot}\,\text{ARROW}}{\tau \to \bot = (\sigma \to \bot) \to \bot}\,\text{FOLD}_r}{\tau \to \bot = \sigma}\,(\text{FOLD}_l/\text{FIX})_x}{\mu\alpha.\,(\alpha \to \bot) = \mu\beta.\,((\beta \to \bot) \to \bot)}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\overbrace{(\mu\alpha.\,(\alpha \to \bot)}^{\equiv:\,\tau} = \overbrace{\mu\beta.\,((\beta \to \bot) \to \bot))}^{\equiv:\,\sigma}{}^{x}}{\tau \to \bot = \sigma}\,\text{UNFOLD}_l}{\tau \to \bot = (\sigma \to \bot) \to \bot}\,\text{UNFOLD}_r \qquad \dfrac{\dfrac{\dfrac{\tau = \sigma \to \bot}{\tau \to \bot = \sigma \to \bot}\,\text{DECOMP}}{(\tau = \sigma)^{x}} \qquad \dfrac{}{\bot = \bot}}{\bot = \bot}\,\text{DECOMP}$$

■

What does the *duality* show? What can it be used for?

- It provides a link between the proof systems $\mathbf{HB}_{\mathbf{0}}^{=}$ and $\mathbf{AK}_{\mathbf{0}}^{=}$ for recursive type equality: It ties together closely the concepts *"consistency w.r.t.* $\mathbf{AK}_{\mathbf{0}}^{=}$*"* and *"derivability in* $\mathbf{HB}_{\mathbf{0}}^{=}$*"*.
- It can help to understand the soundness of the rule ARROW/FIX in $\mathbf{HB}_{\mathbf{0}}^{=}$ along the informal explanation:

  > *"The intuition being that if you cannot find hard evidence proving that the judgement is false then it must be true."* (M. Brandt, [Br97])

  In particular, the duality leads to an alternative soundness proof for $\mathbf{HB}_{\mathbf{0}}^{=}$ by 'reducing' the soundness property of $\mathbf{HB}_{\mathbf{0}}^{=}$ to the property of soundness of $\mathbf{AK}_{\mathbf{0}}^{=}$ w.r.t. $=_{\mu}$ (cf. *extended version* of this paper).

What does the *duality* show? What can it be used for?

- It provides a link between the proof systems $\mathbf{HB}_{\mathbf{0}}^{=}$ and $\mathbf{AK}_{\mathbf{0}}^{=}$ for recursive type equality: It ties together closely the concepts *"consistency w.r.t.* $\mathbf{AK}_{\mathbf{0}}^{=}$*"* and *"derivability in* $\mathbf{HB}_{\mathbf{0}}^{=}$*"*.
- It can help to understand the soundness of the rule ARROW/FIX in $\mathbf{HB}_{\mathbf{0}}^{=}$ along the informal explanation:

  > *"The intuition being that if you cannot find hard evidence proving that the judgement is false then it must be true."* (M. Brandt, [Br97])

  In particular, the duality leads to an alternative soundness proof for $\mathbf{HB}_{\mathbf{0}}^{=}$ by 'reducing' the soundness property of $\mathbf{HB}_{\mathbf{0}}^{=}$ to the property of soundness of $\mathbf{AK}_{\mathbf{0}}^{=}$ w.r.t. $=_{\mu}$ (cf. *extended version* of this paper).
- The soundness proof for $\mathbf{HB}_{\mathbf{0}}^{=}$ can be extended *in principle* to a soundness proof for the axiomatization of $=_{\mu}$ given by Brandt, Henglein (cf. a forthcoming report [CG]).
- *"Non-Result"*: The duality does not also lead to a completeness proof for $\mathbf{HB}_{\mathbf{0}}^{=}$ that is independent from that one given in [BrHe98].

(II)

A Duality in Proof Systems for Bisimulation Equivalence on Cyclic Term Graphs

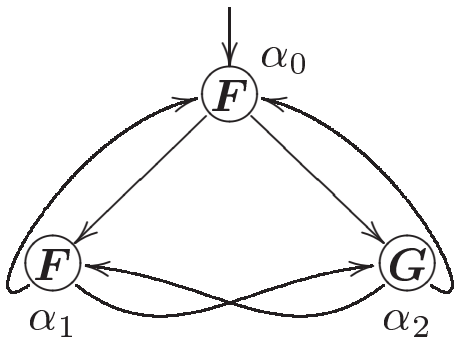
Example 20 (Bisimulation Equivalence on Canonical Term Graph Representations).

We consider the two canonical term graph specifications

$$g := \langle\, \alpha_0 \,|\, E_g \,\rangle := \langle\, \alpha_0 \,|\, \{\alpha_0 = \boldsymbol{F}(\alpha_1, \alpha_2),\ \alpha_1 = \boldsymbol{F}(\alpha_0, \alpha_2),\ \alpha_2 = \boldsymbol{G}(\alpha_1, \alpha_0)\} \,\rangle \quad (1)$$

$$h := \quad (2)$$

in $\mathcal{TGS}(\{F, G\})$. These correspond respectively to the two cyclic term graphs



and

.

■

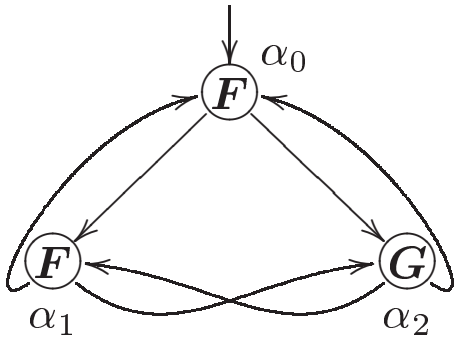
Example 20 (Bisimulation Equivalence on Canonical Term Graph Representations).

We consider the two canonical term graph specifications

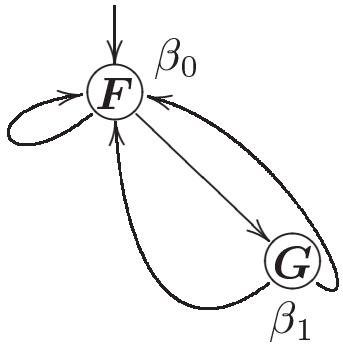
$$g := \langle\, \alpha_0 \,|\, E_g \,\rangle := \langle\, \alpha_0 \,|\, \{\alpha_0 = \boldsymbol{F}(\alpha_1, \alpha_2),\ \alpha_1 = \boldsymbol{F}(\alpha_0, \alpha_2),\ \alpha_2 = \boldsymbol{G}(\alpha_1, \alpha_0)\} \,\rangle \qquad (1)$$

$$h := \langle\, \beta_0 \,|\, E_h \,\rangle := \langle\, \beta_0 \,|\, \{\beta_0 = \boldsymbol{F}(\beta_0, \beta_1),\ \beta_1 = \boldsymbol{G}(\beta_0, \beta_0)\} \,\rangle \qquad (2)$$

in $\mathcal{TGS}(\{F, G\})$. These correspond respectively to the two cyclic term graphs



and



.

■

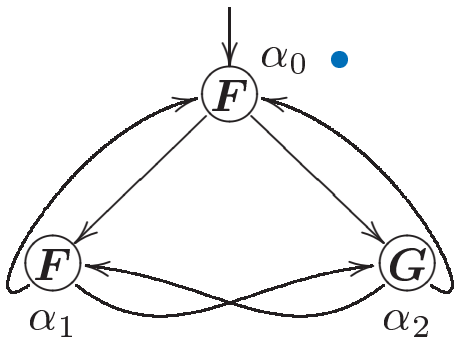
Example 20 (Bisimulation Equivalence on Canonical Term Graph Representations).

We consider the two canonical term graph specifications

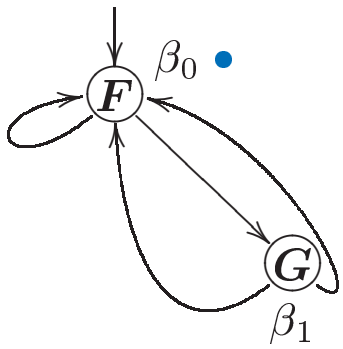
$$g := \langle\, \alpha_0 \,|\, E_g \,\rangle := \langle\, \alpha_0 \,|\, \{\alpha_0 = \boldsymbol{F}(\alpha_1, \alpha_2),\ \alpha_1 = \boldsymbol{F}(\alpha_0, \alpha_2),\ \alpha_2 = \boldsymbol{G}(\alpha_1, \alpha_0)\} \,\rangle \qquad (1)$$

$$h := \langle\, \beta_0 \,|\, E_h \,\rangle := \langle\, \beta_0 \,|\, \{\beta_0 = \boldsymbol{F}(\beta_0, \beta_1),\ \beta_1 = \boldsymbol{G}(\beta_0, \beta_0)\} \,\rangle \qquad (2)$$

in $\mathcal{TGS}(\{F, G\})$. These correspond respectively to the two cyclic term graphs



and



.

$$R := \{(\alpha_0, \beta_0),\qquad\qquad\}$$

■

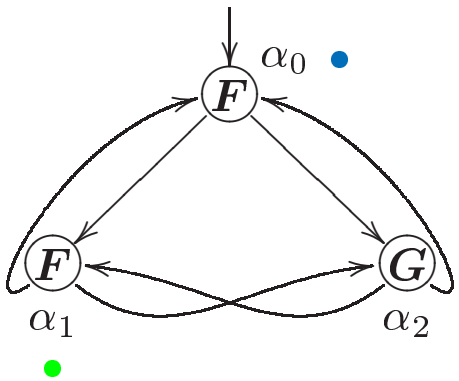
Example 20 (Bisimulation Equivalence on Canonical Term Graph Representations).

We consider the two canonical term graph specifications

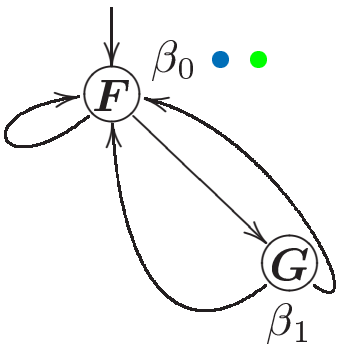
$$g := \langle\, \alpha_0 \,|\, E_g \,\rangle := \langle\, \alpha_0 \,|\, \{\alpha_0 = \boldsymbol{F}(\alpha_1, \alpha_2),\ \alpha_1 = \boldsymbol{F}(\alpha_0, \alpha_2),\ \alpha_2 = \boldsymbol{G}(\alpha_1, \alpha_0)\} \,\rangle \qquad (1)$$

$$h := \langle\, \beta_0 \,|\, E_h \,\rangle := \langle\, \beta_0 \,|\, \{\beta_0 = \boldsymbol{F}(\beta_0, \beta_1),\ \beta_1 = \boldsymbol{G}(\beta_0, \beta_0)\} \,\rangle \qquad (2)$$

in $\mathcal{TGS}(\{F, G\})$. These correspond respectively to the two cyclic term graphs



and



.

$$R := \{(\alpha_0, \beta_0), (\alpha_1, \beta_0), \qquad\}$$

■

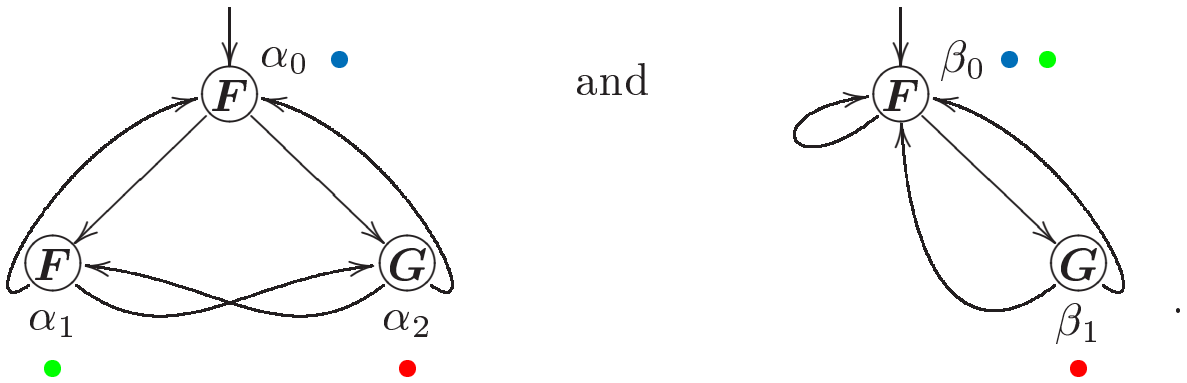
Example 20 (Bisimulation Equivalence on Canonical Term Graph Representations).

We consider the two canonical term graph specifications

$$g := \langle\, \alpha_0 \,|\, E_g \,\rangle := \langle\, \alpha_0 \,|\, \{\alpha_0 = \boldsymbol{F}(\alpha_1, \alpha_2),\ \alpha_1 = \boldsymbol{F}(\alpha_0, \alpha_2),\ \alpha_2 = \boldsymbol{G}(\alpha_1, \alpha_0)\} \,\rangle \qquad (1)$$

$$h := \langle\, \beta_0 \,|\, E_h \,\rangle := \langle\, \beta_0 \,|\, \{\beta_0 = \boldsymbol{F}(\beta_0, \beta_1),\ \beta_1 = \boldsymbol{G}(\beta_0, \beta_0)\} \,\rangle \qquad (2)$$

in $\mathcal{TGS}(\{F, G\})$. These correspond respectively to the two cyclic term graphs



$$R := \{(\alpha_0, \beta_0), (\alpha_1, \beta_0), (\alpha_2, \beta_1)\}$$

■

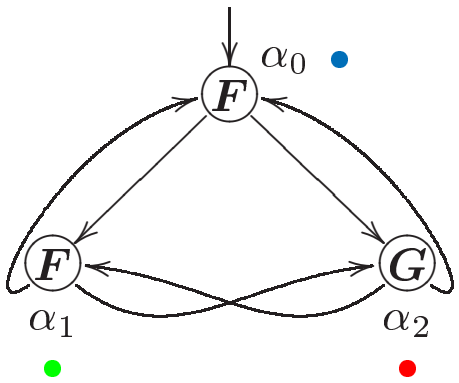
Example 20 (Bisimulation Equivalence on Canonical Term Graph Representations).

We consider the two canonical term graph specifications

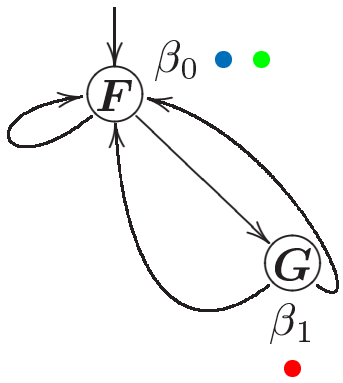
$$g := \langle\, \alpha_0 \,|\, E_g \,\rangle := \langle\, \alpha_0 \,|\, \{\alpha_0 = \boldsymbol{F}(\alpha_1, \alpha_2),\ \alpha_1 = \boldsymbol{F}(\alpha_0, \alpha_2),\ \alpha_2 = \boldsymbol{G}(\alpha_1, \alpha_0)\} \,\rangle \tag{1}$$

$$h := \langle\, \beta_0 \,|\, E_h \,\rangle := \langle\, \beta_0 \,|\, \{\beta_0 = \boldsymbol{F}(\beta_0, \beta_1),\ \beta_1 = \boldsymbol{G}(\beta_0, \beta_0)\} \,\rangle \tag{2}$$

in $\mathcal{TGS}(\{F, G\})$. These correspond respectively to the two cyclic term graphs



and



.

We will speak of the relation $R := \{(\alpha_0, \beta_0), (\alpha_1, \beta_0), (\alpha_2, \beta_1)\}$ as of a *bisimulation* between the canonical term graph specifications g and h (cf. Definition 22). We will write that $g \leftrightarrows h$ holds, i.e. g and h are *bisimilar*. ■

Definition 21. (Canonical Term Graph Specifications). Let Σ be a first-order signature. A *canonical term graph specification* (a *c.t.g.s.*) $g \in \mathcal{TGS}(\Sigma)$ is an equational specification of the form $g = \langle\, \alpha_0 \,|\, \{\alpha_0 = t_0, \ldots, \alpha_n = t_n\} \,\rangle$, where $\alpha_0, \ldots, \alpha_n$ pairwisely different *recursion variables* and for all $i \in \{0, \ldots, n\}$ the terms t_i of the form $F(\alpha_{i1}, \ldots, \alpha_{in_i})$ with some respective function symbol $F \in \Sigma$ of arity n_i and respective variables $\alpha_{i1}, \ldots, \alpha_{in_i} \in \{\alpha_0, \ldots, \alpha_n\}$. ⊠

Definition 21. (Canonical Term Graph Specifications). Let Σ be a first-order signature. A *canonical term graph specification* (a *c.t.g.s.*) $g \in \mathcal{TGS}(\Sigma)$ is an equational specification of the form $g = \langle\, \alpha_0 \,|\, \{\alpha_0 = t_0, \ldots, \alpha_n = t_n\} \,\rangle$, where $\alpha_0, \ldots, \alpha_n$ pairwisely different *recursion variables* and for all $i \in \{0, \ldots, n\}$ the terms t_i of the form $F(\alpha_{i1}, \ldots, \alpha_{in_i})$ with some respective function symbol $F \in \Sigma$ of arity n_i and respective variables $\alpha_{i1}, \ldots, \alpha_{in_i} \in \{\alpha_0, \ldots, \alpha_n\}$. ⊠

Bisimilarity between c.t.g.s.'s is defined in [ArKl95] as follows:

Definition 22. (Bisimulation Equivalence $\leftrightarrows$ on c.t.g.s.'s). Let Σ be a signature. Let $g, h \in \mathcal{TGS}(\Sigma)$, i.e. g and h canonical term graph specifications over Σ of the form $g = \langle\, \alpha_0 \,|\, \{\alpha_0 = t_0, \ldots, \alpha_n = t_n\} \,\rangle$ and $h = \langle\, \alpha'_0 \,|\, \{\alpha'_0 = t'_0, \ldots, \alpha'_{n'} = t'_{n'}\} \,\rangle$.

(a) R is called a *bisimulation* between g and h if and only if

(i) R is a relation with domain $\{\alpha_0, \ldots, \alpha_n\}$ and codomain $\{\alpha'_0, \ldots, \alpha'_{n'}\}$;

(ii) $\alpha_0 \, R \, \alpha'_0$;

(iii) if $\alpha_i \, R \, \alpha'_j$ for some i, j with $0 \leq i \leq n$ and $0 \leq j \leq n'$, and given that $t_i \equiv F(\alpha_{i1}, \ldots, \alpha_{in_i})$ and $t'_j \equiv F'(\alpha'_{j1}, \ldots, \alpha'_{jn'_j})$ with some $n_i, n'_j \in \mathbb{N}_0$, then $F \equiv F'$ (and hence $n_i = n'_j$) and $\alpha_{i1} \, R \, \alpha'_{j1}$, $\ldots$, $\alpha_{in_i} \, R \, \alpha'_{jn'_j}$ must hold.

(b) We say that g and h are *bisimilar* (symbolically denoted by $g \leftrightarrows h$) iff there exists a bisimulation between g and h. ⊠

Definition 23. (Axiomatization $\mathrm{HB}_0^{\leftrightarrows}$ of bisimulation equivalence $\leftrightarrows$ on c.t.g.s.'s analogous to the variant Brandt-Henglein system $\mathrm{HB}_0^{=}$).

Axioms/Marked Assumptions:

$$\text{(REFL)}\ \overline{\langle\, \alpha \,|\, \{\alpha = C, \ldots\}\,\rangle = \langle\, \beta \,|\, \{\beta = C, \ldots\}\,\rangle} \qquad \text{(Assm)}\quad (g = h)^x \ .$$

(if C is a constant symbol in Σ) (with $x \in Mk$)

Rules:

Rule COMP with

$$\frac{\langle\, \alpha_1 \,|\, E_g\,\rangle = \langle\, \beta_1 \,|\, E_h\,\rangle \qquad \ldots \qquad \langle\, \alpha_n \,|\, E_g\,\rangle = \langle\, \beta_n \,|\, E_h\,\rangle}{\langle\, \alpha \,|\, \underbrace{\{\alpha = F(\alpha_1, \ldots, \alpha_n)\} \uplus E_g^{(0)}}_{= E_g}\,\rangle = \langle\, \beta \,|\, \underbrace{\{\beta = F(\beta_1, \ldots, \beta_n)\} \uplus E_h^{(0)}}_{= E_h}\,\rangle}\ \text{COMP}$$

$\boxtimes$

Definition 23. (Axiomatization $\mathrm{HB}_0^{\leftrightarrows}$ of bisimulation equivalence $\leftrightarrows$ on c.t.g.s.'s analogous to the variant Brandt-Henglein system $\mathrm{HB}_0^{=}$).

Axioms/Marked Assumptions:

$$\text{(REFL)}\ \frac{}{\langle\, \alpha \,|\, \{\alpha = C, \ldots\}\,\rangle = \langle\, \beta \,|\, \{\beta = C, \ldots\}\,\rangle}\ (\text{if } C \text{ is a constant symbol in } \Sigma) \qquad \text{(Assm)}\ \ (g = h)^x\ \ (\text{with } x \in Mk)\ .$$

Rules:

Rule COMP and rule COMP/FIX with

$$\frac{\begin{array}{ccc} \{\!\{\, \langle\, \alpha \,|\, E_g \,\rangle = \langle\, \beta \,|\, E_h \,\rangle \,\}\!\}^x & & \{\!\{\, \langle\, \alpha \,|\, E_g \,\rangle = \langle\, \beta \,|\, E_h \,\rangle \,\}\!\}^x \\ \mathcal{D}_1 & & \mathcal{D}_n \\ \langle\, \alpha_1 \,|\, E_g \,\rangle = \langle\, \beta_1 \,|\, E_h \,\rangle & \ldots & \langle\, \alpha_n \,|\, E_g \,\rangle = \langle\, \beta_n \,|\, E_h \,\rangle \end{array}}{\langle\, \alpha \,|\, \underbrace{\{\alpha = F(\alpha_1, \ldots, \alpha_n)\} \uplus E_g^{(0)}}_{= E_g} \,\rangle = \langle\, \beta \,|\, \underbrace{\{\beta = F(\beta_1, \ldots, \beta_n)\} \uplus E_h^{(0)}}_{= E_h} \,\rangle}\ \begin{array}{l}(\text{COMP}/\\ \text{FIX})_x\\ (\text{if s.-cond. } \mathbf{I})\end{array}$$

$\boxtimes$

Definition 23. (Axiomatization $\mathbf{HB}_{\mathbf{0}}^{\leftrightarrows}$ of bisimulation equivalence $\leftrightarrows$ on c.t.g.s.'s analogous to the variant Brandt-Henglein system $\mathbf{HB}_{\mathbf{0}}^{=}$).

Axioms/Marked Assumptions:

$$\text{(REFL)}\ \ \overline{\langle\, \alpha \,|\, \{\alpha = C, \ldots\}\,\rangle = \langle\, \beta \,|\, \{\beta = C, \ldots\}\,\rangle} \quad \text{(if } C \text{ is a constant symbol in } \Sigma\text{)} \qquad \text{(Assm)}\ \ (g = h)^x \ . \quad \text{(with } x \in Mk\text{)}$$

Rules:

Rule COMP and rule COMP/FIX with

$$\frac{\overset{\mathcal{D}_1}{\langle\!\langle\, \langle\, \alpha \,|\, E_g \,\rangle = \langle\, \beta \,|\, E_h \,\rangle \,\rangle\!\rangle^x} \quad \langle\, \alpha_1 \,|\, E_g \,\rangle = \langle\, \beta_1 \,|\, E_h \,\rangle \quad \ldots \quad \overset{\mathcal{D}_n}{\langle\!\langle\, \langle\, \alpha \,|\, E_g \,\rangle = \langle\, \beta \,|\, E_h \,\rangle \,\rangle\!\rangle^x} \quad \langle\, \alpha_n \,|\, E_g \,\rangle = \langle\, \beta_n \,|\, E_h \,\rangle}{\langle\, \alpha \,|\, \underbrace{\{\alpha = F(\alpha_1, \ldots, \alpha_n)\} \uplus E_g^{(0)}}_{= E_g} \,\rangle = \langle\, \beta \,|\, \underbrace{\{\beta = F(\beta_1, \ldots, \beta_n)\} \uplus E_h^{(0)}}_{= E_h} \,\rangle}\ \begin{array}{l}(\text{COMP}/\\ \text{FIX})_x\\ (\text{if s.-cond. } \mathbf{I})\end{array}$$

⊠

Theorem 24 (Soundness and Completeness of $\mathbf{HB}_{\mathbf{0}}^{\leftrightarrows}$ w.r.t. $\leftrightarrows$).
For all canonical term graph specifications $g, h \in \mathcal{TGS}(\Sigma)$ it holds that:

$$\vdash_{\mathbf{HB}_{\mathbf{0}}^{\leftrightarrows}} g = h \quad \Longleftrightarrow \quad g \leftrightarrows h \ .$$

Definition 25. (A 'syntactic matching' proof system $\mathbf{AK}_0^{\leftrightarrows}$ for testing bisimulation equivalence $\leftrightarrows$ on c.t.g.s.'s, which is similar to $\mathbf{AK}_0^{=}$).

Axioms: $\mathbf{AK}_0^{\leftrightarrows}$ possesses **no** *axioms*.

Rules:

$$\frac{\langle\, \alpha \,|\, \overbrace{\{\alpha = F(\alpha_1, \ldots, \alpha_n)\} \uplus E_g^{(0)}}^{= E_g} \,\rangle = \langle\, \beta \,|\, \overbrace{\{\beta = F(\beta_1, \ldots, \beta_n)\} \uplus E_h^{(0)}}^{= E_h} \,\rangle}{\langle\, \alpha_i \,|\, E_g \,\rangle = \langle\, \beta_i \,|\, E_h \,\rangle}\ \text{DECOMP (for } 1 \leq i \leq n)$$

$\boxtimes$

Definition 25. (A 'syntactic matching' proof system $\mathbf{AK}_{\mathbf{0}}^{\leftrightarrows}$ for testing bisimulation equivalence $\underline{\leftrightarrows}$ on c.t.g.s.'s, which is similar to $\mathbf{AK}_{\mathbf{0}}^{=}$).

Axioms: $\mathbf{AK}_{\mathbf{0}}^{\leftrightarrows}$ possesses **no** *axioms*.

Rules:

$$\frac{\langle\, \alpha \,|\, \overbrace{\{\alpha = F(\alpha_1, \ldots, \alpha_n)\} \uplus E_g^{(0)}}^{= E_g} \,\rangle = \langle\, \beta \,|\, \overbrace{\{\beta = F(\beta_1, \ldots, \beta_n)\} \uplus E_h^{(0)}}^{= E_h} \,\rangle}{\langle\, \alpha_i \,|\, E_g \,\rangle = \langle\, \beta_i \,|\, E_h \,\rangle}\ \text{DECOMP (for } 1 \leq i \leq n)$$

⊠

Theorem 27 (Soundness and Completeness of $\mathbf{AK}_{\mathbf{0}}^{\leftrightarrows}$ w.r.t. $\underline{\leftrightarrows}$).
For all canonical term graph specifications g, h it holds that:

$$g = h \text{ is } \mathbf{AK}_{\mathbf{0}}^{\leftrightarrows}\text{-consistent} \iff g \underline{\leftrightarrows} h \ .$$

Definition 25. (A 'syntactic matching' proof system $\mathbf{AK}_0^{\leftrightarrows}$ for testing bisimulation equivalence $\underline{\leftrightarrows}$ on c.t.g.s.'s, which is similar to $\mathbf{AK}_0^{=}$).

Axioms: $\mathbf{AK}_0^{\leftrightarrows}$ possesses **no** *axioms*.

Rules:

$$\frac{\langle\, \alpha \,|\, \overbrace{\{\alpha = F(\alpha_1, \ldots, \alpha_n)\} \uplus E_g^{(0)}}^{= E_g} \,\rangle = \langle\, \beta \,|\, \overbrace{\{\beta = F(\beta_1, \ldots, \beta_n)\} \uplus E_h^{(0)}}^{= E_h} \,\rangle}{\langle\, \alpha_i \,|\, E_g \,\rangle = \langle\, \beta_i \,|\, E_h \,\rangle}\ \text{DECOMP (for } 1 \leq i \leq n)$$

⊠

Definition 26. For all canonical term graph specifications $g, h, \tilde{g}, \tilde{h} \in \mathcal{TGS}(\Sigma)$ with $\tilde{g} = \langle\, \tilde{\alpha} \,|\, \tilde{\alpha} = F(\ldots), \ldots \,\rangle$ and $\tilde{h} = \langle\, \tilde{\beta} \,|\, \tilde{\beta} = G(\ldots), \ldots \,\rangle$ it holds that:

(i) $\tilde{g} = \tilde{h}$ is a *contradiction w.r.t.* $\underline{\leftrightarrows}$ $:\Longleftrightarrow$ $F \not\equiv G$.

(ii) $g = h$ is $\mathbf{AK}_0^{\leftrightarrows}$*-consistent* $:\Longleftrightarrow$ no contradiction w.r.t. $\underline{\leftrightarrows}$ is derivable in $\mathbf{AK}_0^{\leftrightarrows}$ from the assumption $g = h$.

⊠

Theorem 27 (Soundness and Completeness of $\mathbf{AK}_0^{\leftrightarrows}$ w.r.t. $\underline{\leftrightarrows}$).
For all canonical term graph specifications g, h it holds that:

$$g = h \text{ is } \mathbf{AK}_0^{\leftrightarrows}\text{-consistent} \iff g \,\underline{\leftrightarrows}\, h \ .$$

Analogously as in part (I) the following can be defined:

- (Partial) Consistency-Unfoldings in $\mathbf{AK}_{\mathbf{0}}^{\leftrightarrows}$.
- Reflection Mappings $\mathcal{C}(\cdot)$ and $\mathcal{D}(\cdot)$ between p.c.u.'s in $\mathbf{AK}_{\mathbf{0}}^{\leftrightarrows}$ and derivations in $\mathbf{HB}_{\mathbf{0}}^{\leftrightarrows}$.

Analogously as in part (I) the following can be defined:

- (Partial) Consistency-Unfoldings in $\mathbf{AK}_{\mathbf{0}}^{\leftrightarrows}$.
- Reflection Mappings $\mathcal{C}(\cdot)$ and $\mathcal{D}(\cdot)$ between p.c.u.'s in $\mathbf{AK}_{\mathbf{0}}^{\leftrightarrows}$ and derivations in $\mathbf{HB}_{\mathbf{0}}^{\leftrightarrows}$.

Theorem 28. (A Duality between derivations in $\mathbf{HB}_{\mathbf{0}}^{\leftrightarrows}$ and consistency-unfoldings in $\mathbf{AK}_{\mathbf{0}}^{\leftrightarrows}$).

There is a bijective functional relationship between derivations in $\mathbf{HB}_{\mathbf{0}}^{\leftrightarrows}$ *without open assumption classes and consistency-unfoldings in* $\mathbf{AK}_{\mathbf{0}}^{\leftrightarrows}$ *via reflection mappings* $\mathcal{C}(\cdot)$ *and* $\mathcal{D}(\cdot)$*: This means that an analogous statement to that of Theorem 18 is true:*

— *For all canonical term graph specifications* g *and* h*:*

$$\boxed{\begin{array}{c}\tilde{\mathcal{D}}\\ g = h\end{array}} \equiv \mathcal{D}(\tilde{\mathcal{C}}) \qquad \underset{\mathcal{D}(\cdot)}{\overset{\mathcal{C}(\cdot)}{\longmapsto \atop \longleftarrow}} \qquad \mathcal{C}(\tilde{\mathcal{D}}) \equiv \boxed{\begin{array}{c}g = h\\ \tilde{\mathcal{C}}\end{array}}$$

$\mathbf{HB}_{\mathbf{0}}^{\leftrightarrows}$*-derivation of* $g = h$ *without open assumption classes*

consistency-unfolding of $g = h$ *in* $\mathbf{AK}_{\mathbf{0}}^{\leftrightarrows}$

— *The reflection functions* $\mathcal{C}(\cdot)$ *and* $\mathcal{D}(\cdot)$ *are each other's inverses.*

Example 29. (Duality between a derivation in $\mathbf{HB}_0^{\leftrightarrows}$ and a consistency-unfolding in $\mathbf{AK}_0^{\leftrightarrows}$).

$$\dfrac{\dfrac{\dfrac{(\langle\, \alpha_0 \mid E_g \,\rangle = \langle\, \beta_0 \mid E_h \,\rangle)^x}{(\langle\, \alpha_1 \mid E_g \,\rangle = \langle\, \beta_0 \mid E_h \,\rangle)^y} \quad \dfrac{\dfrac{(\langle\, \alpha_2 \mid E_g \,\rangle = \langle\, \beta_1 \mid E_h \,\rangle)^z}{\langle\, \alpha_1 \mid E_g \,\rangle = \langle\, \beta_0 \mid E_h \,\rangle} \quad (\ldots)^x}{(\ldots)^x}}{\dfrac{\langle\, \alpha_2 \mid E_g \,\rangle = \langle\, \beta_1 \mid E_h \,\rangle}{(\langle\, \alpha_1 \mid E_g \,\rangle = \langle\, \beta_0 \mid E_h \,\rangle)^y} \quad (\ldots)^x} \quad \dfrac{(\ldots)^x \quad (\ldots)^z}{(\ldots)^x}}$$

■

Example 29. (Duality between a derivation in $\mathrm{HB}_0^{\leftrightarrows}$ and a consistency-unfolding in $\mathrm{AK}_0^{\leftrightarrows}$).

$$\dfrac{\dfrac{\dfrac{(\langle\,\alpha_1 \mid E_g\,\rangle = \langle\,\beta_0 \mid E_h\,\rangle)^{y} \qquad (\ldots)^{x}}{\langle\,\alpha_2 \mid E_g\,\rangle = \langle\,\beta_1 \mid E_h\,\rangle}\,\mathrm{y} \qquad \dfrac{(\ldots)^{x} \qquad (\ldots)^{z}}{\langle\,\alpha_1 \mid E_g\,\rangle = \langle\,\beta_0 \mid E_h\,\rangle \qquad (\ldots)^{x}}\,\mathrm{z}}{\langle\,\alpha_1 \mid E_g\,\rangle = \langle\,\beta_0 \mid E_h\,\rangle \qquad\qquad \langle\,\alpha_2 \mid E_g\,\rangle = \langle\,\beta_1 \mid E_h\,\rangle}\,\mathrm{x}}{\underbrace{\langle\,\alpha_0 \mid E_g\,\rangle}_{=g \text{ in } (1)} = \underbrace{\langle\,\beta_0 \mid E_h\,\rangle}_{=h \text{ in } (2)}}$$

$$\dfrac{\dfrac{\dfrac{(\langle\,\alpha_0 \mid E_g\,\rangle = \langle\,\beta_0 \mid E_h\,\rangle)^{x}}{(\langle\,\alpha_1 \mid E_g\,\rangle = \langle\,\beta_0 \mid E_h\,\rangle)^{y} \qquad\qquad (\langle\,\alpha_2 \mid E_g\,\rangle = \langle\,\beta_1 \mid E_h\,\rangle)^{z}}}{(\ldots)^{x} \qquad \langle\,\alpha_2 \mid E_g\,\rangle = \langle\,\beta_1 \mid E_h\,\rangle \qquad \langle\,\alpha_1 \mid E_g\,\rangle = \langle\,\beta_0 \mid E_h\,\rangle \qquad (\ldots)^{x}}}{(\langle\,\alpha_1 \mid E_g\,\rangle = \langle\,\beta_0 \mid E_h\,\rangle)^{y} \qquad (\ldots)^{x} \qquad\qquad (\ldots)^{x} \qquad (\ldots)^{z}}$$

■

Summary

(I) A 'duality' in proof systems for Recursive Type Equality.

- The variant system $\mathbf{HB}_{\mathbf{0}}^{=}$ of the axiomatization of *recursive type equality* $=_\mu$ by Brandt and Henglein. A 'syntactic-matching' system $\mathbf{AK}_{\mathbf{0}}^{=}$ à la Ariola and Klop for checking consistency with respect to $=_\mu$.
- Observation of a close connection through "reflection" between a successful consistency-check in $\mathbf{AK}_{\mathbf{0}}^{=}$ and a derivation in $\mathbf{HB}_{\mathbf{0}}^{=}$ in an example.

- *Formalization Step 1:* The conservative extension $\mathbf{e\text{-}HB}_{\mathbf{0}}^{=}$ of $\mathbf{HB}_{\mathbf{0}}^{=}$.
- *Formalization Step 2:* Partial Consistency-Unfoldings and Consistency-Unfoldings in $\mathbf{AK}_{\mathbf{0}}^{=}$.
- *Formalization Step 3:* Reflection Mappings $\mathcal{C}(\cdot)$ and $\mathcal{D}(\cdot)$.
- Main Theorem: A *duality* induced by $\mathcal{C}(\cdot)$ and $\mathcal{D}(\cdot)$ between $\mathbf{e\text{-}HB}_{\mathbf{0}}^{=}$-derivations and consistency-unfoldings in $\mathbf{AK}_{\mathbf{0}}^{=}$.

(II) A 'duality' in proof systems for Bisimulation Equivalence on cyclic term graphs.

Sketch of an analogous *duality theorem* for a similar pair of proof systems $\mathbf{HB}_{\mathbf{0}}^{\leftrightarrows}$ and $\mathbf{AK}_{\mathbf{0}}^{\leftrightarrows}$ concerned with bisimulation equivalence $\leftrightarrows$ on cyclic term graphs.

Issues not treated here. Further Work.

- **With respect to part (I):**
  - Precise *proof-theoretical relationship* between the here considered variant system $\mathbf{HB}_{\mathbf{0}}^{=}$ of the axiomatization $\mathbf{HB}^{=}$ for $=_{\mu}$ introduced by Brandt and Henglein (will be dealt with in a forthcoming report [CG]).
  - The notion of 'consistency-unfoldings' bears an obvious analogy with the concept of *'closed analytic tableau'* as introduced by R. Smullyan.
- **With respect to part (II):**
  - The duality theorem for proof systems w.r.t. $\leftrightarrows$ on c.t.g.s.'s *is stable under the introduction* of rules that formalize *garbage collection* on c.t.g.s.'s like e.g.

$$\frac{\overbrace{\langle\, \alpha \mid E_0 \uplus \{\alpha_i = t_i\}\,\rangle}^{=g} = h}{\langle\, \alpha \mid E_0\,\rangle = h}\ \mathrm{GC}_l \quad (\text{if } \alpha_i \text{ is } \textit{unreachable} \text{ in } g)$$

in a conservative extension of $\mathbf{AK}_{\mathbf{0}}^{\leftrightarrows}$: Cf. the extended version of this paper at

http://www.cs.vu.nl/~clemens/tg02_ext.pdf .

References

[AmCa93] Amadio, R.M., Cardelli, L.: "Subtyping Recursive Types", *ACM Transactions on Programming Languages and Systems 15 (4)*, pp. 575–631, 1993.

[ArKl95] Ariola, Z.M., Klop, J.W.: "Equational Term Graph Rewriting", *Fundamenta Informaticae 26 (3,4)*, pp. 207–240, June 1996; extended version as: *Vrije Universiteit Amsterdam Technical Report IR–391*, September 1995.

[Br97] Brandt, M.: "Recursive Subtyping: Axiomatizations and Computational Interpretations", Master's Thesis, DIKU, Department of Computer Science, University of Copenhagen, August 1997; available on the internet as *TOPPS Technical Report D-352* from:
`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.

[CG] Grabmayer, C.: "Proof-Theoretic Interconnections between Proof Systems for Recursive Type Equality", forthcoming *VU-report*.

Hint at the Proof of Theorem 16. Let $\tau, \sigma \in \mathit{can}\text{-}\mu Tp$. We have to show that

$$\tau = \sigma \text{ is } \mathbf{AK}_{\mathbf{0}}^{=}\text{-consistent} \iff \text{There exists a consistency-unfolding of } \tau = \sigma \text{ in } \mathbf{AK}_{\mathbf{0}}^{=} \ . \tag{3}$$

"$\Leftarrow$": This direction follows by a generalization of the intuitive argumentation explained in the case of Example 10.

"$\Rightarrow$": The implication "$\Rightarrow$" in (3) follows 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 $=_{\mu}$: Suppose that $\tau = \sigma$ is $\mathbf{AK}_{\mathbf{0}}^{=}$-consistent. Then 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; thereby branches in this tree are always cut off 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, because it can be shown that the set of conclusions of derivations from $\tau = \sigma$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ is always finite given that equations that arise from each other by applications of VAR are not counted separately.

□

Alternative Soundness Proof for $\mathbf{HB}_{\mathbf{0}}^{=}$ with respect to $=_{\mu}$

(We mean an alternative soundness proof compared to one that can be built for $\mathbf{HB}_{\mathbf{0}}^{=}$ *in a way as described by Brandt and Henglein in [BrHe98].)*

We proceed by 'reducing' the soundness of $\mathbf{HB}_{\mathbf{0}}^{=}$ w.r.t. $=_{\mu}$ (one half of Theorem 6 to the soundness of $\mathbf{AK}_{\mathbf{0}}^{=}$ w.r.t. $=_{\mu}$ (one half of Theorem 9):

Proof. Suppose that $\vdash_{\mathbf{HB}_{\mathbf{0}}^{=}} \tau = \sigma$ for some $\tau, \sigma \in \mathit{can}\text{-}\mu Tp$, i.e. that $\tau = \sigma$ is a theorem of $\mathbf{HB}_{\mathbf{0}}^{=}$.

This means that there exists a derivation $\tilde{\mathcal{D}}$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ with conclusion $\tau = \sigma$ and without open assumption classes. Let $\tilde{\mathcal{D}}$ be chosen as such a derivation.

Then due to Theorem 18 the reflection $\mathcal{C}(\tilde{\mathcal{D}})$ of $\tilde{\mathcal{D}}$ is a consistency-unfolding of $\tau = \sigma$ in $\mathbf{AK}_{\mathbf{0}}^{=}$.

Hence by Theorem 16 the equation $\tau = \sigma$ is consistent with respect to $\mathbf{AK}_{\mathbf{0}}^{=}$.

And from this Theorem 9, which states the soundness of $\mathbf{AK}_{\mathbf{0}}^{=}$ with respect to $=_{\mu}$, implies that $\tau =_{\mu} \sigma$, i.e. that τ and σ are strongly equivalent. □

Example 30. (Duality between a syntactic-matching tableau in ann-t-$\mathbf{AK}_{\mathbf{0}}^{=}$ and a derivation in e-$\mathbf{HB}_{\mathbf{0}}^{=}$).

<table>
<tr><td colspan="3">$\bullet\ \varepsilon : \overbrace{\mu\alpha.\,(\alpha \to \bot)}^{\equiv:\,\tau} = \overbrace{\mu\beta.\,((\beta \to \bot) \to \bot)}^{\equiv:\,\sigma}$
$\varepsilon : \tau \to \bot = (\sigma \to \bot) \to \bot$</td></tr>
<tr><td colspan="2">$1 : \tau = \sigma \to \bot$
$1 : \tau \to \bot = \sigma \to \bot$</td><td>$2 : \bot = \bot$
X</td></tr>
<tr><td>$\bullet\ 11 : \tau = \sigma$
X</td><td>$12 : \bot = \bot$
X</td><td></td></tr>
</table>

■

Example 30. (Duality between a syntactic-matching tableau in ann-t-$\mathrm{AK}_0^{=}$ and a derivation in e-$\mathrm{HB}_0^{=}$).

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{(\tau = \sigma)^{x}}{\tau \to \bot = \sigma \to \bot}\,\text{ARROW}}{\tau = \sigma \to \bot}}{\tau \to \bot = (\sigma \to \bot) \to \bot}\,\text{FOLD}_r \qquad \dfrac{\dfrac{(\text{REFL})}{\bot = \bot}}{\bot = \bot}\,\text{ARROW}}{\tau \to \bot = \sigma}\,(\text{FOLD}_l/\text{FIX})_{x}}{\mu\alpha.\,(\alpha \to \bot) = \mu\beta.\,((\beta \to \bot) \to \bot)}$$

• $\varepsilon : \overbrace{\mu\alpha.\,(\alpha \to \bot)}^{\equiv:\,\tau} = \overbrace{\mu\beta.\,((\beta \to \bot) \to \bot)}^{\equiv:\,\sigma}$ $\varepsilon : \tau \to \bot = (\sigma \to \bot) \to \bot$	
$1 : \tau = \sigma \to \bot$ $1 : \tau \to \bot = \sigma \to \bot$	$2 : \bot = \bot$ X
• $11 : \tau = \sigma$ X	$12 : \bot = \bot$ X

■