H. Hüttel, C. Stirling: "Actions Speak Louder than Words – Proving Bisimilarity for Context-Free Processes"

Clemens Grabmayer
Department of Computer Science
Vrije Universiteit Amsterdam

26^{th} November 2004

Overview

A. *Introduction and Basics*.

- A decision problem solved by Baeten, Bergstra, and Klop.
- Subsequent developments concerning this problem.
- Recursive BPA-processes. Guardedness, normedness.
- Bisimulation. Greibach normal form, r-GNF. Self-bisimulation.

B. *The tableau decision method by Stirling and Hüttel*.

- The system $\mathbf{HS}^{\sim}$. Tableaux in $\mathbf{HS}^{\sim}$.
- Soundness and Completeness of $\mathbf{HS}^{\sim}$.

C. *"Esoterics": Proof systems for recursive BPA-processes.*

- The proof system $\mathbf{St}^{\sim}$ by Stirling.
- Proof-theoretic relationship between $\mathbf{HS}^{\sim}$ and $\mathbf{St}^{\sim}$.
- A variant system $\mathbf{St}^{\sim}_{\star}$.

A. Introduction and Basics

Introduction (I/II)

The language equality probl. for context-free grammars is *undecidable*.

Baeten, Bergstra, and Klop in [1] adressed the question:

> "Is the equality problem for context-free grammars in Greibach Normal Form solvable when "equality" refers to (a notion corresponding to) bisimulation equivalence?" (1)

They reformulate this as:

> "Is the equality problem for process specifications by means of guarded recursion equations in Basic Process Algebra solvable when "equality" refers to bisimulation equivalence?" (2)

and give a partial solution: the problem referred to in (2) is *decidable* in case that only *normed* process specifications are considered.

Introduction (II/II)

Theorem 1 (Baeten, Bergstra, and Klop, 1987).
Equality of recursively defined normed processes in the graph model of BPA is decidable.

The Proof: is "not easy" (St./Hü. in [5]), "lengthy and impenetrable" (Hü. in [4]);
"relies on isolating a possibly complex periodicity from transition graphs of normed recursively defined BPA-processes" ([5]);
"consists in showing that one can exhibit a decomposition of process graphs with certain regularities" ([4]);
"is based upon the fact that [normed rec. def. BPA-processes] display a very periodical strucure that can be made explicit in the corresponding process graphs (Baeten, Bergstra, Klop in [1])".

(Some Important) Subsequent Developments

- An alternative proof by Caucal (1988, in [2]): reducing the bisimulation problem in question to a decidable rewriting problem (a complete Thue system). (Caucal introduces and uses the notions of “self-bisimulation” and “fundamental relation”.)

- Alternative proof by Stirling and Hüttel in “Actions Speak Louder Than Words . . . ” (1991, in [5]): a tableau decision method for deciding bisimulation equivalence between normed recursively defined BPA-processes.

- Christensen, Hüttel, Stirling (1998,in [3]) proved: bisimulation equivalence is decidable for *all guarded recursive BPA-processes* by adaptations of Caucal’s ideas, not by a tableau decision method.

Contributions made in "Actions Speak Louder. . . "

- A tableau decision method for bisimulation equivalence on normed, recursively defined BPA-processes.

- A sound and complete (sequent-style) proof system for bisim. equiv. on normed, rec. def. BPA-processes ("the theory emanates from 'running the tableau method backwards' "), which extends Milner's axiomatization of regular processes to "context-free" processes.

- Extracting "fundamental relations" out of "successful" tableaux.

(In [4] Hüttel moreover uses the same tableau method to show that *branching bisimulation equivalence* (of Weijland and van Glabbeek) is decidable for normed BPA processes *with silent actions*.)

Recursion Systems and Recursive Specifications

Definition 2. For a set Act of *actions*, and a set $\mathcal{X}$ of variables,

$$E ::= a \mid X \mid E_1 + E_2 \mid E_1 . E_2 \quad (a \in Act,\ X \in \mathcal{X})$$

generates the set of *BPA-(process) expressions*, over Act and $\mathcal{X}$, which we denote by $PExpr(Act, \mathcal{X})$ (or just by $PExpr$).

A *recursion system (in BPA)* is a finite system of equations of the form $\Delta = \{X_1 =_{\text{def}} E_1, \ldots, X_k =_{\text{def}} E_k\}$ such that the X_i are distinct *recursion variables*, and the E_i are BPA-expressions on Act and $\{X_1, \ldots, X_k\}$.

A *recursive specification (in BPA)* is an expression $\langle E \mid \Delta \rangle$ with E a BPA-expression, and Δ is a recursion system such that the variables in E occur among the recursion variables of Δ.

Guardedness

Definition 3. A BPA-expression E is *guarded* if and only if every variable occurrence in E is within the scope of an atomic action.

A recursion system $\Delta = \{X_i =_{\text{def}} E_i \mid 1 \leq i \leq k\}$ in BPA is *guarded* iff all E_i are guarded.

Accordingly, a recursive specification $\langle E \mid \Delta \rangle$ is *guarded* iff the recursion system Δ is guarded.

Example 4. Guarded: $aXYzWX$,

$\{W =_{\text{def}} (aW + b)UWU, U =_{\text{def}} (a + aU)WUW\}$.

Not guarded: $X(a + aW)$, $\langle X \mid X =_{\text{def}} bY, Y =_{\text{def}} (a + X)b \rangle$; however, the latter specification can be rewritten into the guarded form $\langle X \mid X =_{\text{def}} bY, Y =_{\text{def}} (a + bY)b \rangle$.

Convention on Guardedness

We restrict ourselves to guarded recursion systems and recursive specification. Therefore we adopt the following convention.

Convention 5. By referring to a recursion system, or a recursive specification, we actually mean a guarded recursion system, or a guarded recursive specification.

Recursive BPA-Processes

Definition 6 (The LTS generated by a recursion system). A recursion system Δ defines a LTS on $PExpr(Act, \mathcal{X}) \cup \{\epsilon\}$ by the following TSS:

$$\frac{}{a \xrightarrow{a} \epsilon} \text{(if } a \in Act\text{)}$$

$$\frac{E \xrightarrow{a} E'}{E + F \xrightarrow{a} E'} \qquad \frac{F \xrightarrow{a} F'}{E + F \xrightarrow{a} F'} \qquad \frac{E \xrightarrow{a} \epsilon}{E + F \xrightarrow{a} \epsilon} \qquad \frac{F \xrightarrow{a} \epsilon}{E + F \xrightarrow{a} \epsilon}$$

$$\frac{E \xrightarrow{a} E'}{E.F \xrightarrow{a} E'.F} \qquad \frac{E \xrightarrow{a} \epsilon}{E.F \xrightarrow{a} F}$$

$$\frac{E \xrightarrow{a} E'}{X \xrightarrow{a} E'} \text{(if } X =_{\text{def}} E \in \Delta\text{)} \qquad \frac{E \xrightarrow{a} \epsilon}{X \xrightarrow{a} \epsilon} \text{(if } X =_{\text{def}} E \in \Delta\text{)}$$

(Intuitively $E \xrightarrow{a} E'$ means $\langle E \mid \Delta \rangle \xrightarrow{a} \langle E' \mid \Delta \rangle$).

By the transition $E \xrightarrow{a} E'$ we denote the statement that the formula $E \xrightarrow{a} E'$ is derivable from the above TSS.

We write $E \xrightarrow{a}_{(\Delta)} E'$ for a transition $E \xrightarrow{a} E'$ if the underlying recursion system Δ has to be emphasized.

By $E \xrightarrow{w} E'$ we mean $E \xrightarrow{a_1} E_1 \xrightarrow{a_2} \ldots \xrightarrow{a_{n-1}} E_{n-1} \xrightarrow{a_n} E'$ holds for some $E_1, \ldots, E_{n-1}$, given that $w \in Act^+$ with $w = a_1 \ldots a_n$.

Languages and Traces

Definition 7. Let Δ be a recursion system. For all BPA-expressions E, the language $L(E)$ accepted by E, and the set $Tr(E)$ of (finite) traces for E are defined by

$$L(E) =_{\text{def}} \left\{ w \in Act^{+} \mid E \xrightarrow{w} \epsilon \right\},$$

$$Tr(E) =_{\text{def}} \left\{ w \in Act^{+} \mid E \xrightarrow{w} \epsilon \;\vee\; (\exists E' \in PExpr)\,[\, E \xrightarrow{w} E' \,] \right\}.$$

Theorem 8. *Language equivalence is undecidable for normed recursive specifications in BPA: the problem of deciding, for normed recursion systems Δ and expressions E and F in BPA, whether $L(E) = L(F)$ holds is undecidable.*

Corollary 9. *Trace equivalence is undecidable for normed recursive specifications in BPA.*

Normedness (I/II)

Definition 10. Let Δ be a recursion system. The *norm* of a BPA-expression E is defined as

$$\|E\| =_{\text{def}} \min\left\{|w| \;\middle|\; E \xrightarrow{w}_{(\Delta)} \epsilon,\ w \in Act^{+}\right\} \in (\omega \setminus \{0\}) \cup \{\infty\}\,.$$

Δ is said to be *normed* if and only if, for all recursion variables X of Δ, $\|X\| < \infty$ holds. If Δ is normed, then the maximal norm of a recursion variable of Δ is defined by

$$m_{\Delta} =_{\text{def}} \max\left\{\|X\| \mid X \in RVar(\Delta)\right\} \in \omega \setminus \{0\}\,.$$

A recursive specification $\langle E \mid \Sigma \rangle$ is *normed* if and only if Σ is normed.

Normedness (II/II)

Proposition 11.

(i) It is (easily) decidable whether a recursion system Δ *is normed or not.*

(ii) Let Δ *be a recursion system. Then for all normed BPA-expressions it holds:*

$$\|E+F\| = \min\{\|E\|, \|F\|\}\,, \quad \|E.F\| = \|E\| + \|F\|\,.$$

Example 12. The guarded recursion system

$$\{\, X =_{\mathsf{def}} a(Y+ZX)+aXb\,,\quad Y =_{\mathsf{def}} aZ(Y+bXXX)+aZ\,,\quad Z =_{\mathsf{def}} a\,\}$$

is normed, and it holds: $\|X\| = 3$, $\|Y\| = 2$, and $\|Z\| = 1$.

Bisimulation (w.r.t. two recursion systems)

Definition 13. Let Δ_1, Δ_2 be recursion systems on Act and $\mathcal{X}$; we let $PExpr =_{\text{def}} PExpr(Act, \mathcal{X})$.

A binary relation R on $PExpr$ is called a *bisimulation with respect to Δ_1 and Δ_2* iff for all $E, E', F, F' \in PExpr$ and $a \in Act$:

1. $E \, R \, F \;\;\&\;\; E \xrightarrow{a}_{(\Delta_1)} E' \;\;\Rightarrow\;\; (\exists F') \left[F \xrightarrow{a}_{(\Delta_2)} F' \;\;\&\;\; E' \, R \, F' \right]$;
2. $E \, R \, F \;\;\&\;\; F \xrightarrow{a}_{(\Delta_2)} F' \;\;\Rightarrow\;\; (\exists E') \left[E \xrightarrow{a}_{(\Delta_1)} E' \;\;\&\;\; E' \, R \, F' \right]$;
3. $E \, R \, F \;\;\&\;\; E \xrightarrow{a}_{(\Delta_1)} \epsilon \;\;\Rightarrow\;\; F \xrightarrow{a}_{(\Delta_2)} \epsilon$;
4. $E \, R \, F \;\;\&\;\; F \xrightarrow{a}_{(\Delta_2)} \epsilon \;\;\Rightarrow\;\; E \xrightarrow{a}_{(\Delta_1)} \epsilon$;

if $E \, R \, F$, we write $E \sim_{\Delta_1, \Delta_2} F$ (intuitively: $\langle E \mid \Delta_1 \rangle \overset{(R)}{\sim} \langle F \mid \Delta_2 \rangle$).

Bisimulation (w.r.t. two/one recursion systems)

Definition 13 (Continued). We let

$$\sim_{\Delta_1,\Delta_2} \;=_{\text{def}}\; \left\{ \langle E, F\rangle \in PExpr^2 \;\middle|\; \begin{array}{l} E\,R\,F \text{ holds for some} \\ \text{bisimulation } R \text{ w.r.t. } \Delta_1,\, \Delta_2 \end{array} \right\}.$$

Let $\langle E_1 \,|\, \Delta_1\rangle$, $\langle E_2 \,|\, \Delta_2\rangle$ be recursive specifications. We say that $\langle E_1 \,|\, \Delta_1\rangle$ and $\langle E_2 \,|\, \Delta_2\rangle$ are bisimilar (notation $\langle E_1 \,|\, \Delta_1\rangle \sim \langle E_2 \,|\, \Delta_2\rangle$) if and only if $E_1 \sim_{\Delta_1,\Delta_2} E_2$.

Let Δ be a recursion system. By a *bisimulation with respect to* Δ we mean a bisimulation with respect to Δ and Δ; and we let

$$\sim_{\Delta} \;=_{\text{def}}\; \sim_{\Delta,\Delta} \;.$$

Bisimulation (w.r.t. one recursion systems)

Proposition 14. *Let* Δ *be a recursion systems on* Act *and* $\mathcal{X}$.

A binary relation R *on* $PExpr$ *is a* bisimulation with respect to Δ *iff for all* $E, E', F, F' \in PExpr$ *and* $a \in Act$*:*

1. $E \, R \, F \;\;\&\;\; E \xrightarrow{a}_{(\Delta)} E' \;\;\Rightarrow\;\; (\exists F') \left[F \xrightarrow{a}_{(\Delta)} F' \;\;\&\;\; E' \, R \, F' \right]$;
2. $E \, R \, F \;\;\&\;\; F \xrightarrow{a}_{(\Delta)} F' \;\;\Rightarrow\;\; (\exists E') \left[E \xrightarrow{a}_{(\Delta)} E' \;\;\&\;\; E' \, R \, F' \right]$;
3. $E \, R \, F \;\;\&\;\; E \xrightarrow{a}_{(\Delta)} \epsilon \;\;\Rightarrow\;\; F \xrightarrow{a}_{(\Delta)} \epsilon$;
4. $E \, R \, F \;\;\&\;\; F \xrightarrow{a}_{(\Delta)} \epsilon \;\;\Rightarrow\;\; E \xrightarrow{a}_{(\Delta)} \epsilon$;

if $E \, R \, F$, *we write* $E \sim_{\Delta} F$ *(intuitively:* $\langle E \mid \Delta \rangle \overset{(R)}{\sim} \langle F \mid \Delta \rangle$*).*

Greibach normal form

Definition 15. Let $\Delta = \{X_1 =_{\text{def}} E_1, \ldots, X_n =_{\text{def}} E_n\}$ be a recursion system. Δ is in *Greibach normal form (GNF)* if and only if, for all $i \in \{1, \ldots, n\}$,

$$X_i =_{\text{def}} \sum_{j=1}^{n_i} a_{ij}\alpha_{ij} \quad \text{for some } n_i \in \omega \backslash \{0\}, \text{ and, for all } 1 \leq i \leq n_i,\ a_i \in Act \text{ and } \alpha_i \in \mathcal{X}^*. \tag{3}$$

And we say that, for $k \in \omega \backslash \{0\}$, Δ is in *k-GNF* if and only if, for all $i \in \{1, \ldots, n\}$, (1) holds with $|\alpha_{ij}| < k$. 3-GNF is also called *restricted Greibach normal form (r-GNF)*.

A recursive specification $\langle E \mid \Delta \rangle$ is in *GNF* (in *k-GNF*, in *r-GNF*) if and only if Δ is in GNF (in k-GNF, in r-GNF).

Transformation to r-GNF (I/II)

Theorem 16. *Every guarded recursive specification* $\langle E \,|\, \Delta \rangle$ *can effectively be transformed into a recursive specification* $\langle E \,|\, \Delta' \rangle$ *in r-GNF such that* $\langle E \,|\, \Delta \rangle \sim \langle E \,|\, \Delta' \rangle$*, and such that* $\langle E \,|\, \Delta' \rangle$ *is normed iff* $\langle E \,|\, \Delta \rangle$ *is normed.*

Or equivalently: every guarded recursion system Δ *can effectively be transformed into a recursion system* Δ' *in r-GNF such that:*

- Δ' *is in r-GNF;*
- $RVar(\Delta) \subseteq RVar(\Delta')$*, and for all* $X \in RVar(\Delta)$ *it holds:* $X \sim_{\Delta,\Delta'} X$*;*
- Δ' *is normed if and only if* Δ *is normed.*

Transformation to r-GNF (II/II)

Example 17. The normed guarded recursion system Δ

$$\Delta = \{\, X =_{\text{def}} a(Y + ZX) + aXb\,, \quad Y =_{\text{def}} aZ(Y + bXXX) + aZ\,, \quad Z =_{\text{def}} a \,\}$$

can be transformed into the normed recursion system Δ' in r-GNF:

$$\begin{aligned}
\Delta' = \{\, X &=_{\text{def}} aX_{Y+ZX} + aXX_b\,, \\
Y &=_{\text{def}} aZX_{Y+X_bXXX} + aZ\,, \\
Z &=_{\text{def}} a\,, \\
X_b &=_{\text{def}} b\,, \\
X_{Y+ZX} &=_{\text{def}} aZX_{Y+X_bXXX} + aZ + aX\,, \\
X_{Y+X_bXXX} &=_{\text{def}} aZX_{Y+X_bXXX} + aZ + bU_{XX}X\,, x \\
U_{XX} &=_{\text{def}} aX + aU_{XX_b}X\,, \\
U_{XX_b} &=_{\text{def}} bX \,\}
\end{aligned}$$

Transformation to r-GNF (II/II)

Example 17. The normed guarded recursion system Δ

$$\Delta = \{\, X =_{\text{def}} a(Y + ZX) + aXb\,,\quad Y =_{\text{def}} aZ(Y + bXXX) + aZ\,,\quad Z =_{\text{def}} a \,\}$$

can be transformed into the normed recursion system Δ' in r-GNF:

$$\begin{aligned} \Delta' = \{\, X &=_{\text{def}} aX_2 + aXX_1\,, \\ Y &=_{\text{def}} aZX_3 + aZ\,, \\ Z &=_{\text{def}} a\,, \\ X_1 &=_{\text{def}} b\,, \\ X_2 &=_{\text{def}} aZX_3 + aZ + aX\,, \\ X_3 &=_{\text{def}} aZX_3 + aZ + bU_1X\,, \\ U_1 &=_{\text{def}} aX + aU_2X\,, \\ U_2 &=_{\text{def}} bX \,\} \end{aligned}$$

Self-bisimulation (w.r.t. a recursion systems) (I/II)

For a binary relation R on $PExpr$, we denote by $\underset{R}{\leftrightarrow}^{*}$ the *least congruence relation w.r.t. sequential composition*.

Definition 18 (Caucal). Let Δ be a recursion systems on Act and $\mathcal{X}$. A binary relation R on $PExpr$ is a *self-bisimulation with respect to* Δ iff for all $E, E', F, F' \in PExpr$ and $a \in Act$:

Self-bisimulation (w.r.t. a recursion systems) (I/II)

For a binary relation R on $PExpr$, we denote by $\underset{R}{\leftrightarrow}^{*}$ the *least congruence relation w.r.t. sequential composition*.

Definition 18 (Caucal). Let Δ be a recursion systems on Act and $\mathcal{X}$. A binary relation R on $PExpr$ is a *self-bisimulation with respect to* Δ iff for all $E, E', F, F' \in PExpr$ and $a \in Act$:

1. $E\, R\, F \;\;\&\;\; E \xrightarrow{a}_{(\Delta)} E' \;\;\Rightarrow\;\; (\exists F')\left[F \xrightarrow{a}_{(\Delta)} F' \;\;\&\;\; E' \underset{R}{\leftrightarrow}^{*} F'\right]$;
2. $E\, R\, F \;\;\&\;\; F \xrightarrow{a}_{(\Delta)} F' \;\;\Rightarrow\;\; (\exists E')\left[E \xrightarrow{a}_{(\Delta)} E' \;\;\&\;\; E' \underset{R}{\leftrightarrow}^{*} F'\right]$;

Self-bisimulation (w.r.t. a recursion systems) (I/II)

For a binary relation R on $PExpr$, we denote by $\underset{R}{\leftrightarrow}^{*}$ the *least congruence relation w.r.t. sequential composition*.

Definition 18 (Caucal). Let Δ be a recursion systems on Act and $\mathcal{X}$. A binary relation R on $PExpr$ is a *self-bisimulation with respect to* Δ iff for all $E, E', F, F' \in PExpr$ and $a \in Act$:

1. $E\, R\, F \;\;\&\;\; E \xrightarrow{a}_{(\Delta)} E' \;\;\Rightarrow\;\; (\exists F')\left[F \xrightarrow{a}_{(\Delta)} F' \;\;\&\;\; E' \underset{R}{\leftrightarrow}^{*} F'\right]$;
2. $E\, R\, F \;\;\&\;\; F \xrightarrow{a}_{(\Delta)} F' \;\;\Rightarrow\;\; (\exists E')\left[E \xrightarrow{a}_{(\Delta)} E' \;\;\&\;\; E' \underset{R}{\leftrightarrow}^{*} F'\right]$;
3. $E\, R\, F \;\;\&\;\; E \xrightarrow{a}_{(\Delta)} \epsilon \;\;\Rightarrow\;\; F \xrightarrow{a}_{(\Delta)} \epsilon$;
4. $E\, R\, F \;\;\&\;\; F \xrightarrow{a}_{(\Delta)} \epsilon \;\;\Rightarrow\;\; E \xrightarrow{a}_{(\Delta)} \epsilon$.

Self Bisimulation (w.r.t. a recursion systems) (II/II)

Lemma 19 (Caucal). *Let Δ be a recursion system.*

If R is a self-bisimulation with respect to Δ, then $\underset{R}{\leftrightarrow}^* \subseteq \sim_{\Delta}$.

Self Bisimulation (w.r.t. a recursion systems) (II/II)

Lemma 19 (Caucal). *Let* Δ *be a recursion system.*
If R *is a self-bisimulation with respect to* Δ*, then* $\underset{R}{\leftrightarrow}^{*} \subseteq \sim_{\Delta}$.

Corollary 20. *Let* Δ *be a recursion system.*
For all $E, F \in PExpr$ *it holds:*

$$E \sim_{\Delta} F \iff (\exists R \textit{ self-bisimulation}) \left[E \, R \, F \right] .$$

B. The tableau decision method by Stirling/Hüttel

Tableau System HS$^{\sim}(\Delta)$ by Hüttel/Stirling (I/II)

Rules in **HS**$^{\sim}(\Delta)$ within subtableaux:

$$\frac{X\alpha = Y\beta}{E\alpha = F\beta}\,\text{REC} \qquad (\text{if } X =_{\text{def}} E \text{ and } Y =_{\text{def}} F \text{ are in } \Delta)$$

$$\frac{a\alpha = a\beta}{\alpha = \beta}\,\text{PREFIX}$$

$$\frac{(\sum_{i=1}^{n} a_i\alpha_i)\alpha = (\sum_{i=1}^{m} b_i\beta_i)\beta}{\{a_i\alpha_i\alpha = b_{f(i)}\beta_{f(i)}\beta\}_{i=1}^{n} \qquad \{a_{g(j)}\alpha_{g(j)}\alpha = b_j\beta_j\beta\}_{j=1}^{m}}\,\text{SUM}$$

where $n, m \geq 1$ and $f : \{1, \ldots, n\} \to \{1, \ldots, m\}$
and $g : \{1, \ldots, m\} \to \{1, \ldots, n\}$

Tableau System HS$^{\sim}(\Delta)$ by Hüttel/Stirling (II/II)

Rules in **HS**$^{\sim}(\Delta)$ for new subtableaux :

$$\frac{\alpha_i\alpha = \beta_i\beta}{\alpha_i\gamma = \beta_i}\text{SUBL} \qquad \text{(if } \alpha = \gamma\beta \text{ is the residual)}$$

$$\frac{\alpha_i\alpha = \beta_i\beta}{\alpha_i = \beta_i\gamma}\text{SUBR} \qquad \text{(if } \gamma\alpha = \beta \text{ is the residual)}$$

A tableau in $\mathbf{HS}^{\sim}(\Delta)$

Example 21. Given the recursion system

$$\Delta = \{X =_{\mathsf{def}} aYX + b,\; Y =_{\mathsf{def}} bX,\; A =_{\mathsf{def}} aC + b,\; C =_{\mathsf{def}} bAA\}\ ,$$

the following is a *successful* tableau in $\mathbf{HS}^{\sim}(\Delta)$:

$$\dfrac{X = A}{\dfrac{aYX + b = aC + b}{\dfrac{aYX = aC}{\dfrac{YX = C}{\dfrac{YX = C}{\dfrac{bXX = bAA}{\dfrac{XX = AA}{\dfrac{(aYX + b)X = (aC + b)A}{\dfrac{aYXX = aCA}{\dfrac{YXX = CA}{YX = C}\,\mathsf{SUBR}}\,\mathsf{PREFIX} \qquad \dfrac{bX = bA}{X = A}\,\mathsf{PREFIX}}\,\mathsf{SUM}}\,\mathsf{REC}}\,\mathsf{PREFIX}}\,\mathsf{REC}}\,\mathsf{SUB}}\,\mathsf{PREFIX} \qquad \dfrac{b = b}{\epsilon = \epsilon}\,\mathsf{PREFIX}}\,\mathsf{SUM}}\,\mathsf{REC}$$

Soundness and Completeness of $\mathsf{HS}^{\sim}(\Delta)$, Decidability of $\sim$

Theorem 22. *Let Δ be a normed recursion system that is in r-GNF. Let $\mathcal{X} = RVar(\Delta)$.*

Then for all $X, Y \in \mathcal{X}$ and $\alpha, \beta \in \mathcal{X}^*$ it holds that:

$$(\exists \mathcal{T}) \begin{bmatrix} \mathcal{T} \textit{ is successful tableaux} \\ \textit{for } X\alpha = Y\beta \textit{ in } \mathsf{HS}^{\sim}(\Delta) \end{bmatrix} \iff X\alpha \sim_{\Delta} Y\beta\,.$$

Theorem 23 (Decidability of $\sim$). *The problem of deciding, for a normed recursion system Δ in r-GNF, and for $X, Y \in \mathcal{X}$, $\alpha, \beta \in \mathcal{X}^*$, whether $X\alpha \sim_{\Delta} Y\beta$ holds, is decidable.*

Applicable for not normed processes? – Sometimes:

Example 24. Given the recursion system

$$\Delta = \{X =_{\text{def}} aX,\ Y =_{\text{def}} aZ,\ Z =_{\text{def}} aY\}\ ,$$

the following is a *successful* tableau in $\mathbf{HS}^{\sim}(\Delta)$:

$$\dfrac{\dfrac{\dfrac{\dfrac{X = Y}{aX = aZ}\,\text{REC}}{X = Z}\,\text{SUM, PREFIX}}{aX = aY}\,\text{REC}}{X = Y}\,\text{SUM, PREFIX}$$

Applicable for not normed processes? – Not always:

Example 25. Given the recursion system

$$\Delta = \{X =_{\text{def}} aX,\ Y =_{\text{def}} aYY\}\ ,$$

the following is an infinite, *not successful* tableau in $\mathbf{HS}^{\sim}(\Delta)$:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{X = Y}{aX = aYY}\,\text{REC}}{X = YY}\,\text{SUM, PREFIX}}{aX = (aY)YY}\,\text{REC}}{X = YYY}\,\text{SUM, PREFIX}}{aX = (aYY)YY}\,\text{REC}}{X = YYYY}\,\text{SUM, PREFIX}$$

$$\vdots$$

$$\vdots$$

C. Proof Systems for Recursive BPA-Processes.

The proof system $\mathsf{St}^{\sim}(\Delta)$ by Stirling (I/II)

Possible open (marked) assumptions:

(R11) $\quad (X\alpha = Y\beta)^u \quad$ (where $X, Y \in \mathcal{X}$ and $\alpha, \beta \in \mathcal{X}^*$)

Equivalence and *Congruence*:

$$\text{R1}\ \frac{}{E = E}\ \text{REFL} \qquad \text{R2}\ \frac{\begin{array}{c}\mathcal{D}_1\\ E = F\end{array}}{F = E}\ \text{SYMM} \qquad \text{R3}\ \frac{\begin{array}{c}\mathcal{D}_1\\ E = F\end{array} \quad \begin{array}{c}\mathcal{D}_2\\ F = G\end{array}}{F = G}\ \text{TRANS}$$

$$\text{R4}\ \frac{\begin{array}{c}\mathcal{D}_1\\ E_1 = F_1\end{array} \qquad \begin{array}{c}\mathcal{D}_2\\ E_2 = F_2\end{array}}{E_1 + E_2 = F_1 + F_2}\ + \qquad \text{R5}\ \frac{\begin{array}{c}\mathcal{D}_1\\ E_1 = F_1\end{array} \qquad \begin{array}{c}\mathcal{D}_2\\ E_2 = F_2\end{array}}{E_1 E_2 = F_1 F_2}\ \cdot$$

The proof system $\mathrm{St}^{\sim}(\Delta)$ by Stirling (II/II)

BPA-axioms:

$$\text{R6}\;\frac{}{E+F=F+E} \qquad \text{R7}\;\frac{}{(E+F)+G=E+(F+G)}$$

$$\text{R8}\;\frac{}{E+E=E} \qquad \text{R9}\;\frac{}{(E+F)G=EG+FG}$$

$$\text{R10}\;\frac{}{(EF)G=E(FG)}$$

Recursion/Fixpoint:

$$\begin{array}{c}[X\alpha = Y\beta]^{u}\\ \mathcal{D}_1\end{array}$$

$$\text{R12}\;\frac{E\alpha = F\beta}{X\alpha = Y\beta}\,\text{REC}^{-1}/\text{FIX},\ u \qquad (\text{if } X =_{\text{def}} E\,,\ Y =_{\text{def}} F \text{ in } \Delta)$$

A comparable inference rule

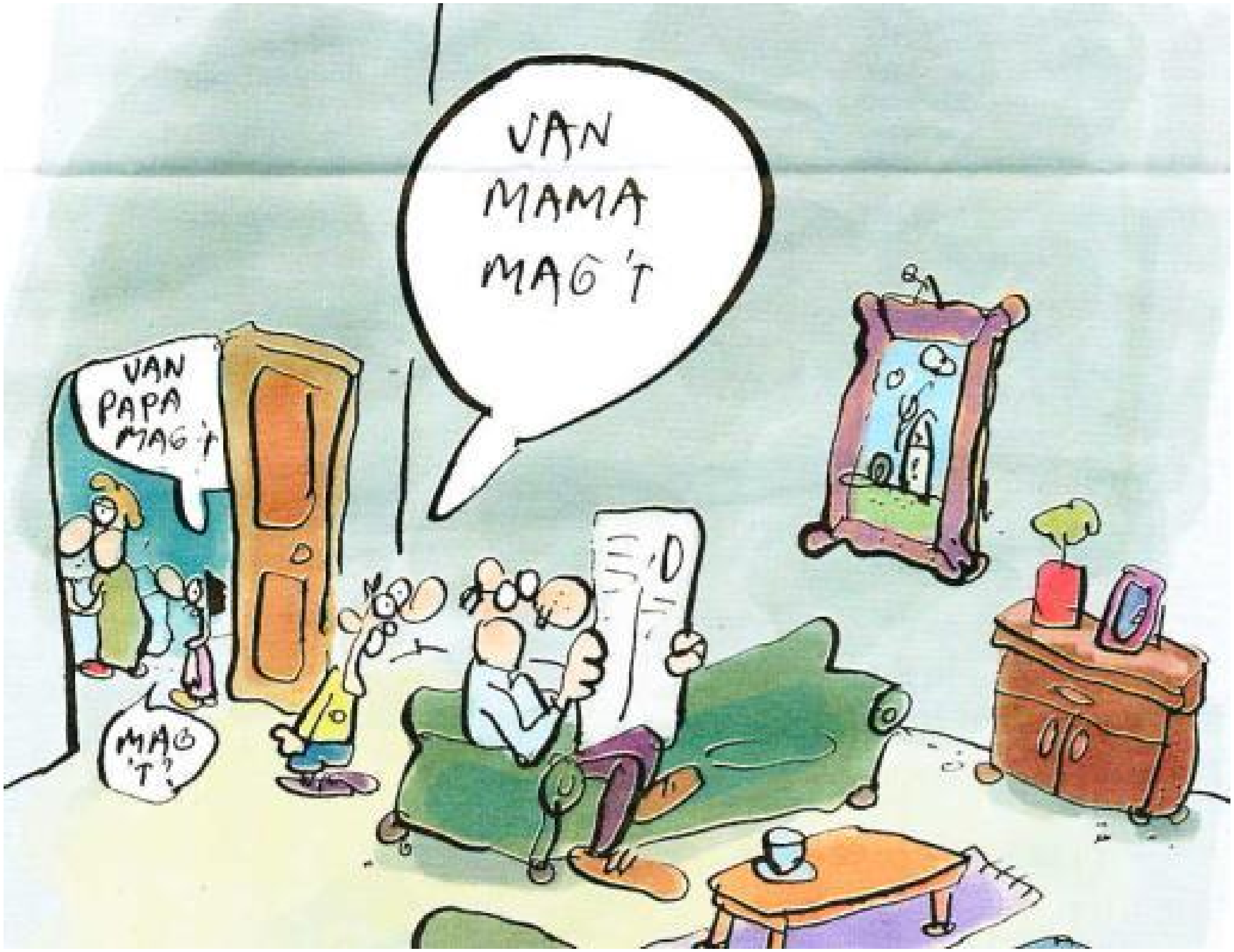
The rule REC^{-1}/FIX in **St**$^{\sim}(\Delta)$ is comparable to the rule ARROW/FIX in an axiomatization of "recursive type equality" by Brandt and Henglein (1998). This rule enables applications of the form:

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

Parental Advisory: Explicit Trap

Parent trap. België 1, 21.40 uur.

Parental Advisory: Explicit Trap



A Circularity between Shamanism and Science

SA 6.54 1.10.2004 MA 19.42
SU 18.34 MU 10.06
275 - 91 Theresia v. K. J. 40. Woche

Der Medizinmann sagt zu den Indianern: „Ich glaube der Winter wird sehr streng. Sammelt viel Brennholz." Zur Sicherheit ruft er am nächsten Tag bei der meteorologischen Station an und fragt: „Wie wird denn der Winter?" „Sicher sehr streng. Die Indianer sammeln Brennholz wie verrückt."

A Circularity between Shamanism and Science

SA 6.54	1.10.2004	MA 19.42
SU 18.34		MU 10.06
275 - 91	Theresia v. K. J.	40. Woche

Der Medizinmann sagt zu den Indianern: „Ich glaube der Winter wird sehr streng. Sammelt viel Brennholz." Zur Sicherheit ruft er am nächsten Tag bei der meteorologischen Station an und fragt: „Wie wird denn der Winter?" „Sicher sehr streng. Die Indianer sammeln Brennholz wie verrückt."

The medicine man tells his fellow natives: "I expect the coming winter to be very rigorous. Go out and collect a lot of firewood."

To be on the safe side, he calls the meteorogical station the next day. "How's the winter going to get?" he asks. "Very rigorous, for sure. The natives are collecting firewood like mad," he is told.

"Asynchronous Coinductive Unfolding"?

Let us consider the rule $\text{REC}_l^{-1}/\text{FIX}$ with applications of the forms

$$\begin{array}{c} [X\alpha = F]^{u} \\ \mathcal{D}_1 \\ \dfrac{E\alpha = F}{X\alpha = F}\,\text{REC}_l^{-1}/\text{FIX},\ u \end{array}$$

for all $X =_{\text{def}} E$ is in Δ

"Asynchronous Coinductive Unfolding"?

Let us consider the rules REC_l^{-1}/FIX and REC_r^{-1}/FIX with applications of the forms

$$\dfrac{\begin{array}{c}[X\alpha = F]^{u} \\ \mathcal{D}_1 \\ E\alpha = F\end{array}}{X\alpha = F}\,\text{REC}_l^{-1}/\text{FIX},\ u \qquad\qquad \dfrac{\begin{array}{c}[E = Y\beta]^{u} \\ \mathcal{D}_1 \\ E = F\beta\end{array}}{E = Y\beta}\,\text{REC}_r^{-1}/\text{FIX},\ u$$

for all $X =_{\text{def}} E$ is in Δ, and respectively, for all $Y =_{\text{def}} F$ in Δ.

"Asynchronous Coinductive Unfolding"?

Let us consider the rules $\mathsf{REC}_l^{-1}/\mathsf{FIX}$ and $\mathsf{REC}_r^{-1}/\mathsf{FIX}$ with applications of the forms

$$\dfrac{\begin{matrix}[X\alpha = F]^{u} \\ \mathcal{D}_1 \\ E\alpha = F\end{matrix}}{X\alpha = F}\,\mathsf{REC}_l^{-1}/\mathsf{FIX},\ u \qquad\qquad \dfrac{\begin{matrix}[E = Y\beta]^{u} \\ \mathcal{D}_1 \\ E = F\beta\end{matrix}}{E = Y\beta}\,\mathsf{REC}_r^{-1}/\mathsf{FIX},\ u$$

for all $X =_{\text{def}} E$ is in Δ, and respectively, for all $Y =_{\text{def}} F$ in Δ.

Question: Can the rule $\mathsf{REC}^{-1}/\mathsf{FIX}$ in $\mathbf{St}^{\sim}(\Delta)$ be replaced by one or both of the rules above with the result of an equivalent theory?

"Asynchronous Coinductive Unfolding"?

Let us consider the rules $\mathsf{REC}_l^{-1}/\mathsf{FIX}$ and $\mathsf{REC}_r^{-1}/\mathsf{FIX}$ with applications of the forms

$$\dfrac{\begin{array}{c}[X\alpha = F]^{u} \\ \mathcal{D}_1 \\ E\alpha = F\end{array}}{X\alpha = F}\,\mathsf{REC}_l^{-1}/\mathsf{FIX},\ u \qquad\qquad \dfrac{\begin{array}{c}[E = Y\beta]^{u} \\ \mathcal{D}_1 \\ E = F\beta\end{array}}{E = Y\beta}\,\mathsf{REC}_r^{-1}/\mathsf{FIX},\ u$$

for all $X =_{\mathsf{def}} E$ is in Δ, and respectively, for all $Y =_{\mathsf{def}} F$ in Δ.

Question: Can the rule $\mathsf{REC}^{-1}/\mathsf{FIX}$ in $\mathbf{St}^{\sim}(\Delta)$ be replaced by one or both of the rules above with the result of an equivalent theory?

Answer: No. Removing $\mathsf{REC}^{-1}/\mathsf{FIX}$ from $\mathbf{St}^{\sim}(\Delta)$ and adding any of $\mathsf{REC}_{l/r}^{-1}/\mathsf{FIX}$ leads to an ext. of $\mathbf{St}^{\sim}(\Delta)$ that is unsound w.r.t. $\sim_{\Delta}$.

"Asynchronous Coinductive Unfolding" is unsound

Let $\mathcal{S}$ be the result of removing the rule $\mathsf{REC}^{-1}/\mathsf{FIX}$ from $\mathbf{St}^{\sim}(\Delta)$ but adding at least one of the rules $\mathsf{REC}_{l/r}^{-1}/\mathsf{FIX}$.

It is easy to see that $\mathcal{S}$ is an extension of $\mathbf{St}^{\sim}(\Delta)$.

"Asynchronous Coinductive Unfolding" is unsound

Let $\mathcal{S}$ be the result of removing the rule REC^{-1}/FIX from $\mathbf{St}^{\sim}(\Delta)$ but adding at least one of the rules $\mathsf{REC}_{l/r}^{-1}$/FIX.

It is easy to see that $\mathcal{S}$ is an extension of $\mathbf{St}^{\sim}(\Delta)$. However

$$\vdash_{\mathcal{S}} E = F \quad \Longrightarrow \quad E \sim_{\Delta} F$$

does not hold for all $E, F \in PExpr$:

“Asynchronous Coinductive Unfolding” is unsound

Let $\mathcal{S}$ be the result of removing the rule $\mathsf{REC}^{-1}/\mathsf{FIX}$ from $\mathbf{St}^{\sim}(\Delta)$ but adding at least one of the rules $\mathsf{REC}_{l/r}^{-1}/\mathsf{FIX}$.

It is easy to see that $\mathcal{S}$ is an extension of $\mathbf{St}^{\sim}(\Delta)$. However

$$\vdash_{\mathcal{S}} E = F \quad \Longrightarrow \quad E \sim_{\Delta} F$$

does not hold for all $E, F \in PExpr$: For all $(X =_{\mathsf{def}} E) \in \Delta$, $\alpha \in \mathcal{X}^*$, and $F \in PExpr$,

$$\dfrac{\dfrac{\dfrac{\dfrac{}{E\alpha = E\alpha}\,\mathsf{REFL}}{X\alpha = E\alpha}\,\mathsf{REC}_l^{-1}/\mathsf{FIX}}{E\alpha = X\alpha}\,\mathsf{SYMM} \qquad (X\alpha = F)^{u}}{\dfrac{E\alpha = F}{X\alpha = F}\,\mathsf{REC}_l^{-1}/\mathsf{FIX},\ u}\,\mathsf{TRANS}$$

is a derivation in $\mathcal{S}$. For given $(X =_{\mathsf{def}} E) \in \Delta$, $X\alpha \sim_{\Delta} F$ will obviously not hold for all process expressions $F \in PExpr$ and $\alpha \in \mathcal{X}^*$.

Soundness and Completeness of $\mathbf{St}^{\sim}(\Delta)$

Theorem 26 (Hüttel/Stirling, '93). *Let Δ be a normed recursion system in BPA with set $\mathcal{X}$ of recursion variables.*

Then for all $X, Y \in \mathcal{X}$ and $\alpha, \beta \in \mathcal{X}^*$ it holds that

$$\vdash_{\mathbf{St}^{\sim}(\Delta)} X\alpha = Y\beta \quad \Longleftrightarrow \quad X\alpha \sim_{\Delta} Y\beta \, .$$

A Derivation in $\mathbf{St}^{\sim}(\Delta)$

Example 27. Given the recursion system

$$\Delta = \{X = aYX + b,\ Y = bX,\ A = aC + b,\ C = bAA\}\ ,$$

the following is a derivation in $\mathbf{St}^{\sim}(\Delta)$:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{a=a}\,\text{R1} \quad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{a=a}\,\text{R1} \quad \dfrac{\dfrac{(YX=C)^{v} \quad \dfrac{}{X=X}\,\text{R1}}{YXX=CX}\,\text{R5} \quad \dfrac{\dfrac{}{C=C}\,\text{R1} \quad (X=A)^{u}}{CX=CA}\,\text{R5}}{YXX=CA}\,\text{R3}}{aYXX=aCX}\,\text{R5} \quad \dfrac{\dfrac{}{b=b}\,\text{R1} \quad (X=A)^{u}}{bX=bA}\,\text{R5}}{aYXX+bX=aCA+bA}\,\text{R4}}{(aYX+b)X=(aC+b)A}\,\text{R9, R3, R2}}{XX=AA}\,\text{R12}}{}}{}}{}\quad\dfrac{}{b=b}}{bXX=bAA}\,\text{R5}}{YX=C}\,\text{R12, } v}{aYX=aC}\,\text{R5} \quad \dfrac{}{b=b}\,\text{R1}}{aYX+b=aC+b}\,\text{R4}}{X=A}\,\text{R12, } u$$

Proof-th. Relation betw. HS$^{\sim}(\Delta)$ and St$^{\sim}(\Delta)$ (I/II)

Example 28. There is a close correspondence between the tableau in **HS**$^{\sim}(\Delta)$ from Example 21 and the proof in **St**$^{\sim}(\Delta)$ from Example 27:

$$
\dfrac{
  \dfrac{
    \dfrac{
      \dfrac{
        \dfrac{
          \dfrac{
            \dfrac{
              \overline{a = a}
              \qquad
              \dfrac{
                \dfrac{(YX = C)^{v} \quad \overline{X = X}}{YXX = CX}\,\cdot
                \quad
                \dfrac{\overline{C = C} \quad (X = A)^{u}}{CX = CA}\,\cdot
              }{YXX = CA}\,\text{TRANS}
            }{aYXX = aCX}\,\cdot
            \qquad
            \dfrac{\overline{b = b} \quad (X = A)^{u}}{bX = bA}\,\cdot
          }{aYXX + bX = aCA + bA}\,+
        }{(aYX + b)X = (aC + b)A}\,\text{BPA-Ax's, TRANS, SYMM}
      }{XX = AA}\,\text{REC}^{-1}
      \quad \overline{b = b}
    }{bXX = bAA}\,\cdot
  }{YX = C}\,\text{REC}^{-1}/\text{FIX}, v
  \quad \overline{a = a}
}{aYX = aC}\,\cdot
$$

$$
\dfrac{\dfrac{aYX = aC \qquad \overline{b = b}}{aYX + b = aC + b}\,+}{X = A}\,\text{REC}^{-1}/\text{FIX}, \boldsymbol{u}
$$

$$
\dfrac{(X = A)^{u}}{aYX + b = aC + b}\,\text{REC}
$$

$$
\dfrac{aYX + b = aC + b}{
  \dfrac{
    \dfrac{
      \dfrac{
        \dfrac{aYX = aC}{YX = C}\,\text{PREFIX}
      }{(YX = C)^{v}}\,\text{SUB}
    }{bXX = bAA}\,\text{REC}
  }{XX = AA}\,\text{PREFIX}
  \qquad
  \dfrac{b = b}{\epsilon = \epsilon}\,\text{PREFIX}
}\,\text{SUM}
$$

$$
\dfrac{XX = AA}{(aYX + b)X = (aC + b)A}\,\text{REC}
$$

$$
\dfrac{(aYX + b)X = (aC + b)A}{
  \dfrac{\dfrac{aYXX = aCA}{YXX = CA}\,\text{PREFIX}}{(YX = C)^{v}}\,\text{SUBR}
  \qquad
  \dfrac{bX = bA}{(X = A)^{u}}\,\text{PREFIX}
}\,\text{SUM}
$$

Proof-th. Relation betw. HS$^{\sim}(\Delta)$ and St$^{\sim}(\Delta)$ (II/II)

By admitting rules $a.$ (for all $a \in Act$) that are derivable in **HS**$^{\sim}(\Delta)$ this correspondence becomes even closer:

$$\dfrac{(YX=C)^{v} \quad \overline{X=X}}{YXX=CX}\,\cdot \qquad \dfrac{\overline{C=C} \quad (X=A)^{u}}{CX=CA}\,\cdot$$

$$\dfrac{YXX=CX \quad CX=CA}{YXX=CA}\;\text{TRANS}$$

$$\dfrac{YXX=CA}{aYXX=aCX}\;a. \qquad \dfrac{(X=A)^{u}}{bX=bA}\;b.$$

$$\dfrac{aYXX=aCX \quad bX=bA}{aYXX+bX=aCA+bA}\;+$$

$$\dfrac{aYXX+bX=aCA+bA}{(aYX+b)X=(aC+b)A}\;\text{BPA-Ax's, TRANS, SYMM}$$

$$\dfrac{(aYX+b)X=(aC+b)A}{XX=AA}\;\text{REC}^{-1}$$

$$\dfrac{XX=AA}{bXX=bAA}\;b. \qquad \dfrac{bXX=bAA}{YX=C}\;\text{REC}^{-1}/\text{FIX},\ v$$

$$\dfrac{YX=C}{aYX=aC}\;a. \qquad \dfrac{aYX=aC \quad \overline{b=b}}{aYX+b=aC+b}\;+$$

$$\dfrac{aYX+b=aC+b}{X=A}\;\text{REC}^{-1}/\text{FIX},\ u$$

$$\dfrac{(X=A)^{u}}{aYX+b=aC+b}\;\text{REC}$$

$$\dfrac{aYX+b=aC+b}{aYX=aC \qquad b=b}\;\text{SUM} \qquad \dfrac{b=b}{\epsilon=\epsilon}\;\text{PREFIX}$$

$$\dfrac{aYX=aC}{YX=C}\;\text{PREFIX} \qquad \dfrac{YX=C}{(YX=C)^{v}}\;\text{SUB}$$

$$\dfrac{(YX=C)^{v}}{bXX=bAA}\;\text{REC} \qquad \dfrac{bXX=bAA}{XX=AA}\;\text{PREFIX}$$

$$\dfrac{XX=AA}{(aYX+b)X=(aC+b)A}\;\text{REC}$$

$$\dfrac{(aYX+b)X=(aC+b)A}{aYXX=aCA \qquad bX=bA}\;\text{SUM}$$

$$\dfrac{aYXX=aCA}{YXX=CA}\;\text{PREFIX} \qquad \dfrac{YXX=CA}{(YX=C)^{v}}\;\text{SUBR} \qquad \dfrac{bX=bA}{(X=A)^{u}}\;\text{PREFIX}$$

A duality between derivations in BH and ‘consistency-unfoldings’ in AK

Example 29. Duality between a proof of $alt = zip(zeros, ones)$ in **BH** a consistency-unfolding in **AK** :

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\overline{0=0} \qquad \dfrac{\dfrac{\overline{1=1} \qquad \big(alt = zip(zeros, ones)\big)^{u}}{1 : alt = 1 : zip(zeros, ones)}\,\text{COMP}}{1 : alt = zip(1 : ones, zeros)}}{0 : 1 : alt = 0 : zip(ones, zeros)}\,\text{COMP}}{0 : 1 : alt = zip(0 : zeros, ones)}}{alt = zip(0 : zeros, ones)}}{alt = zip(zeros, ones)}\,\text{FOLD}_r/\text{FIX},\ u
$$

$$
\begin{array}{c}
alt = zip(zeros, ones) \\ \hline
alt = zip(0 : zeros, ones) \\ \hline
0 : 1 : alt = zip(0 : zeros, ones) \\ \hline
0 : 1 : alt = 0 : zip(ones, zeros) \\ \hline
0 = 0 \qquad 1 : alt = zip(1 : ones, zeros) \quad \text{DECOMP} \\ \hline
1 : alt = 1 : zip(zeros, ones) \\ \hline
1 = 1 \qquad alt = zip(zeros, ones) \quad \text{DECOMP}
\end{array}
$$

Looping back to the top

A duality between $\mathbf{HB}_{\mathbf{0}}^{=}$ and $\mathbf{AK}_{\mathbf{0}}^{=}$

Example 30. A duality between a derivation in $\mathbf{HB}_{\mathbf{0}}^{=}$ and a *consistency-unfolding* in $\mathbf{AK}_{\mathbf{0}}^{=}$:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{(\tau \to \bot = (\sigma \to \bot) \to \bot)^{u}}{\tau = \sigma}\,\text{FOLD}_{l/r} \qquad \dfrac{}{\bot = \bot}\,\text{(REFL)}}{\tau \to \bot = \sigma \to \bot}\,\text{ARROW}}{\tau = \sigma \to \bot}\,\text{FOLD}_{l} \qquad \dfrac{}{\bot = \bot}\,\text{(REFL)}}{\tau \to \bot = (\sigma \to \bot) \to \bot}\,\text{ARROW/FIX}, u}{\underbrace{\mu\alpha.\,(\alpha \to \bot)}_{\equiv \tau} = \underbrace{\mu\beta.\,((\beta \to \bot) \to \bot)}_{\equiv \sigma}}\,\text{FOLD}_{l/r}$$

$$\mu\alpha.\,(\alpha \to \bot) = \mu\beta.\,((\beta \to \bot) \to \bot)$$
$$\text{UNFOLD}_{l/r}:\quad (\tau \to \bot = (\sigma \to \bot) \to \bot)^{u}$$
$$\text{DECOMP}:\quad \tau = \sigma \to \bot \qquad \bot = \bot$$
$$\text{UNFOLD}_{l}:\quad \tau \to \bot = \sigma \to \bot$$
$$\text{DECOMP}:\quad \tau = \sigma \qquad \bot = \bot$$
$$\text{UNFOLD}_{l/r}:\quad (\tau \to \bot = (\sigma \to \bot) \to \bot)^{u}$$

The variant system $\mathbf{St}^{\sim}_{\star}(\Delta)$ of $\mathbf{St}^{\sim}(\Delta)$ (I/II)

Possible open (marked) assumptions:

(Assm) $(E_1 E_2 = F_1 F_2)^u$

Equivalence and *Congruence*:

$$\frac{}{E = E}\,\text{REFL} \qquad \frac{\begin{array}{c}\mathcal{D}_1\\ E = F\end{array}}{F = E}\,\text{SYMM} \qquad \frac{\begin{array}{c}\mathcal{D}_1\\ E = F\end{array} \quad \begin{array}{c}\mathcal{D}_2\\ F = G\end{array}}{E = G}\,\text{TRANS}$$

$$\text{R4}\,\frac{\begin{array}{c}\mathcal{D}_1\\ E_1 = F_1\end{array} \qquad \begin{array}{c}\mathcal{D}_2\\ E_2 = F_2\end{array}}{E_1 + E_2 = F_1 + F_2}\,+$$

$$\text{R5}\,\frac{\begin{array}{c}\mathcal{D}_1\\ E_1 = F_1\end{array} \qquad \begin{array}{c}[E_1 E_2 = F_1 F_2]^u\\ \mathcal{D}_2\\ E_2 = F_2\end{array}}{E_1 E_2 = F_1 F_2}\,\cdot/\text{FIX},\, u$$

The Variant System $\mathsf{St}^{\sim}_{\star}(\Delta)$ of $\mathsf{St}^{\sim}(\Delta)$ (II/II)

BPA-axioms:

$$\frac{}{E + F = F + E}\,\mathsf{A1} \qquad \frac{}{(E + F) + G = E + (F + G)}\,\mathsf{A2}$$

$$\frac{}{E + E = E}\,\mathsf{A3} \qquad \frac{}{(E + F)G = EG + FG}\,\mathsf{A4}$$

$$\frac{}{(EF)G = E(FG)}\,\mathsf{A5}$$

Recursion:

$$\frac{}{X = E}\,\mathsf{REC} \qquad (\text{if } X =_{\text{def}} E \text{ is in } \Delta)$$

A Derivation in $\mathbf{St}_\star^{\sim}(\Delta)$

Example 31. Given the recursion system

$$\Delta =_{\text{def}} \{X = aYX + b,\ Y = bX,\ A = aC + b,\ C = bAA\}\ ,$$

the following is a derivation in $\mathbf{St}_\star^{\sim}(\Delta)$:

$$\dfrac{(bXX = bAA)^{v}}{YX = C}\ (\Longrightarrow) \qquad \dfrac{YX = C \qquad \overline{X = X}}{YXX = CX}\ \cdot$$

$$\dfrac{(aYX = aC)^{u} \qquad \overline{b = b}}{aYX + b = aC + b}\ + \qquad \dfrac{aYX + b = aC + b}{X = A}\ (\Longrightarrow) \qquad \dfrac{\overline{C = C} \qquad X = A}{CX = CA}\ \cdot$$

$$\dfrac{YXX = CX \qquad CX = CA}{YXX = CA}\ \text{TRANS} \qquad \dfrac{\overline{a = a} \qquad YXX = CA}{aYXX = aCX}\ \cdot$$

$$\dfrac{(aYX = aC)^{u} \qquad \overline{b = b}}{aYX + b = aC + b}\ + \qquad \dfrac{aYX + b = aC + b}{X = A}\ (\Longrightarrow) \qquad \dfrac{\overline{b = b} \qquad X = A}{bX = bA}\ \cdot$$

$$\dfrac{aYXX = aCX \qquad bX = bA}{aYXX + bX = aCA + bA}\ +$$

$$\dfrac{aYXX + bX = aCA + bA}{(aYX + b)X = (aC + b)A}\ (\Longrightarrow)\ \text{BPA-Ax's, SYMM, TRANS}$$

$$\dfrac{(aYX + b)X = (aC + b)A}{XX = AA}\ (\Longrightarrow)\ \text{FIX},\ v \qquad \dfrac{\overline{b = b} \qquad XX = AA}{bXX = bAA}\ \cdot$$

$$\dfrac{bXX = bAA}{YX = C}\ (\Longrightarrow)\ \text{FIX},\ u \qquad \dfrac{\overline{a = a} \qquad YX = C}{aYX = aC}\ \cdot$$

$$\dfrac{aYX = aC \qquad \overline{b = b}}{aYX + b = aC + b}\ + \qquad \dfrac{aYX + b = aC + b}{X = A}\ (\Longrightarrow)\ \text{REC}^{-1}\text{, SYMM, TRANS}$$

Soundness and Completeness of $\mathbf{St}_{\star}^{\sim}(\Delta)$

Theorem 32. *Let Δ be a normed recursion system in BPA with set $\mathcal{X}$ of recursion variables.*

Then for all BPA-expressions E and F with variables in $\mathcal{X}$ it holds that

$$\vdash_{\mathbf{St}_{\star}^{\sim}(\Delta)} E = F \quad \Longleftrightarrow \quad E \sim_{\Delta} F \; .$$

Recursive Process Expressions

Definition 33. For a set Act of actions, the $RPExpr(Act)$ of *recursive process expressions (in BPA)* on Act is generated by:

$$\boldsymbol{p} ::= a \mid \langle X \mid \Delta \rangle \mid \boldsymbol{p}_1 + \boldsymbol{p}_2 \mid \boldsymbol{p}_1 . \boldsymbol{p}_2$$

$(a \in Act,\ \langle X \mid \Delta \rangle$ BPA-process specification)

A recursive process expression $\boldsymbol{p}$ is *guarded* (or *normed*) if and only if all BPA-process expressions occurring in $\boldsymbol{p}$ are guarded (normed).

We denote by $gRPExpr(Act)$ and by $gnRPExpr(Act)$ the set of recursive process expressions that are guarded, and respectively, guarded and normed. Often we let the set Act be implicit and use the denotations $RPExpr$, $gRPExpr$, and $gnRPExpr$ for $RPExpr(Act)$, $gRPExpr(Act)$, and $gnRPExpr(Act)$.

The system $\mathrm{St}_{\star}^{\sim}$ (I/II)

Possible open assumptions:

(Assm) $(\boldsymbol{p}_1\boldsymbol{p}_2 = \boldsymbol{q}_1\boldsymbol{q}_2)^{u}$

Equivalence and *Congruence*:

$$\frac{}{\boldsymbol{p} = \boldsymbol{p}}\,\mathsf{REFL} \qquad \frac{\begin{array}{c}\mathcal{D}_1\\ \boldsymbol{p} = \boldsymbol{q}\end{array}}{\boldsymbol{q} = \boldsymbol{p}}\,\mathsf{SYMM} \qquad \frac{\begin{array}{c}\mathcal{D}_1\\ \boldsymbol{p} = \boldsymbol{r}\end{array} \quad \begin{array}{c}\mathcal{D}_2\\ \boldsymbol{r} = \boldsymbol{q}\end{array}}{\boldsymbol{p} = \boldsymbol{q}}\,\mathsf{TRANS}$$

$$\mathsf{R4}\,\frac{\begin{array}{c}\mathcal{D}_1\\ \boldsymbol{p}_1 = \boldsymbol{q}_1\end{array} \qquad \begin{array}{c}\mathcal{D}_2\\ \boldsymbol{p}_2 = \boldsymbol{q}_2\end{array}}{\boldsymbol{p}_1 + \boldsymbol{p}_2 = \boldsymbol{q}_1 + \boldsymbol{q}_2}\,+ \qquad \mathsf{R5}\,\frac{\begin{array}{c}\mathcal{D}_1\\ \boldsymbol{p}_1 = \boldsymbol{q}_1\end{array} \qquad \begin{array}{c}[\boldsymbol{p}_1\boldsymbol{p}_2 = \boldsymbol{q}_1\boldsymbol{q}_2]^{u}\\ \mathcal{D}_2\\ \boldsymbol{p}_2 = \boldsymbol{q}_2\end{array}}{\boldsymbol{p}_1\boldsymbol{p}_2 = \boldsymbol{q}_1\boldsymbol{q}_2}\,\cdot/\mathsf{FIX},\, u$$

The system $\mathsf{St}_{\star}^{\sim}$ (II/II)

BPA-axioms:

$$\frac{}{\boldsymbol{p}+\boldsymbol{q}=\boldsymbol{q}+\boldsymbol{p}}\text{A1} \qquad \frac{}{(\boldsymbol{p}+\boldsymbol{q})+\boldsymbol{r}=\boldsymbol{p}+(\boldsymbol{q}+\boldsymbol{r})}\text{A2}$$

$$\frac{}{\boldsymbol{p}+\boldsymbol{p}=\boldsymbol{p}}\text{A3} \qquad \frac{}{(\boldsymbol{p}+\boldsymbol{q})\boldsymbol{r}=\boldsymbol{p}\boldsymbol{r}+\boldsymbol{q}\boldsymbol{r}}\text{A4}$$

$$\frac{}{(\boldsymbol{p}\boldsymbol{q})\boldsymbol{r}=\boldsymbol{p}(\boldsymbol{q}\boldsymbol{r})}\text{A5}$$

Recursive Definition Principle (RDP):

$$\frac{}{\langle X_i \mid \Delta\rangle = E_i(\langle X_i \mid \Delta\rangle, \ldots, \langle X_n \mid \Delta\rangle)}\text{RDP}$$

(for all $n \in \omega \setminus \{0\}$, $1 \le i \le n$, and recursion systems Δ of the form $\Delta = \{X_1 =_{\mathsf{def}} E_1(X_1, \ldots, X_n),\ \ldots,\ X_n =_{\mathsf{def}} E_n(X_1, \ldots, X_n)\}$)

Soundness and Completeness of $\mathsf{St}^{\sim}$

Theorem 34. $\mathsf{St}^{\sim}_{\star}$ *is sound with respect to guarded recursive process expressions; that is, for all* $\boldsymbol{p}_1, \boldsymbol{p}_2 \in gRPExpr$*, it holds that:*

$$\vdash_{\mathsf{St}^{\sim}_{\star}} \boldsymbol{p}_1 = \boldsymbol{p}_2 \quad \Longrightarrow \quad \boldsymbol{p}_1 \sim \boldsymbol{p}_2 \ .$$

Theorem 35. $\mathsf{St}^{\sim}_{\star}$ *is sound and complete with respect to normed guarded recursive process expressions; that is, for all* $\boldsymbol{p}_1, \boldsymbol{p}_2 \in gnRPExpr$*, it holds that:*

$$\vdash_{\mathsf{St}^{\sim}_{\star}} \boldsymbol{p}_1 = \boldsymbol{p}_2 \quad \Longleftrightarrow \quad \boldsymbol{p}_1 \sim \boldsymbol{p}_2 \ .$$

Full Circle

I. *Introduction and Basics*.

- The decision problem solved by Baeten, Bergstra, and Klop.
- Subsequent develpments concerning this problem.
- Recursive BPA-processes. Guardedness, normedness.
- Bisimulation. Self-bisimulation. Greibach normal form, r-GNF.

II. *The tableau decision method by Stirling and Hüttel*.

- The system $\mathbf{HS}^{\sim}(\Delta)$. Tableaux in $\mathbf{HS}^{\sim}(\Delta)$.
- Soundness and Completeness of $\mathbf{HS}^{\sim}(\Delta)$.

III. *"Esoterics"*. Proof systems for recursive BPA-processes.

- The proof system $\mathbf{St}^{\sim}(\Delta)$ by Stirling.
- Proof-theoretic relationship between $\mathbf{HS}^{\sim}(\Delta)$ and $\mathbf{St}^{\sim}(\Delta)$.
- Variant systems $\mathbf{St}^{\sim}_{\star}(\Delta)$ and $\mathbf{St}^{\sim}_{\star}$.

References

[1] Baeten, J.C.M., Bergstra, J.A., Klop, J.W.: "Decidability of Bisimulation Equivalence for Processes Generating Context-Free Languages", *JACM* **40**:3 (1993) 653–682.

[2] Caucal, D.: "Graphes canoniques de graphes algébriques", *Informatique théorique et Applications (RAIRO)* **24**:4 (1990) 339–352.

[3] Christensen, S., Hüttel, H., Stirling, C.: "Bisimilarity is Decidable for All Context-Free Processes", *Proc. of CONCUR '92* (1992).

[4] Hüttel, H.: "Decidability, Behavioural Equivalences and Infinite Transition Graphs", Ph.D. thesis, Department of Computer Science, Edinburgh University (1991).

[5] Hüttel, H., Stirling, C.: "Actions Speak Louder Than Words: Proving Bisimilarity for Context-Free Processes", *Proceedings of LICS '91*, IEEE Computer Society Press (1991) 376–386.

A Duality between Derivations in BH and 'Consistency-Unfoldings' in AK (I/II)

Definition. We define the stream terms $zeros$, $ones$, and alt, as well as the operation zip on stream terms by

$$zeros =_{\text{def}} 0 : zeros \; ,$$

$$ones =_{\text{def}} 1 : ones \; ,$$

$$alt =_{\text{def}} 0 : 1 : alt \; ,$$

$$zip(a : s,\, t) =_{\text{def}} a : zip(t,\, s) \quad \text{(for all stream terms } s,\, t) \; .$$

A Duality between Derivations in BH and ‘Consistency-Unfoldings’ in AK (III/II)

Example. A *consistency-Unfolding* in a proof system à la Ariola/Klop of the equation $alt = zip(zeros, ones)$:

$$\dfrac{\dfrac{\dfrac{\dfrac{alt = zip(zeros, ones)}{alt = zip(0 : zeros, ones)}}{0 : 1 : alt = zip(0 : zeros, ones)}}{0 : 1 : alt = 0 : zip(ones, zeros)}}{0 = 0 \qquad \dfrac{\dfrac{1 : alt = zip(1 : ones, zeros)}{1 : alt = 1 : zip(zeros, ones)}}{1 = 1 \qquad alt = zip(zeros, ones)}\ \text{DECOMP}}\ \text{DECOMP}$$

Looping back to the top

A Duality between Derivations in BH and ‘Consistency-Unfoldings’ in AK (IV/II)

Example. A *proof* of the equation $alt = zip(zeros, ones)$ in a proof system à la Brandt-Henglein:

$$
\dfrac{
  \dfrac{
    \dfrac{
      \dfrac{
        \dfrac{}{0 = 0}
        \qquad
        \dfrac{
          \dfrac{
            \dfrac{}{1 = 1} \qquad \big(alt = zip(zeros, ones)\big)^{u}
          }{1 : alt = 1 : zip(zeros, ones)}\ \text{COMP}
        }{1 : alt = zip(1 : ones, zeros)}
      }{0 : 1 : alt = 0 : zip(ones, zeros)}\ \text{COMP}
    }{0 : 1 : alt = zip(0 : zeros, ones)}
  }{alt = zip(0 : zeros, ones)}
}{alt = zip(zeros, ones)}\ \text{FOLD}_r/\text{FIX},\ u
$$