Regular Expressions Under the Process Interpretation

Clemens Grabmayer[1]

(Partly) Joint work with
Jos Baeten and Flavio Corradini

[1] Department of Computer Science
Vrije Universiteit Amsterdam
clemens@cs.vu.nl
http://www.cs.vu.nl/~clemens

Workshop on Proof Theory and Rewriting
Obergurgl, September 5–9, 2006

Overview

Overview

The Language Interpretation L (Kleene)

$$0 \xmapsto{L} \text{empty set } \emptyset$$

$$1 \xmapsto{L} \{\lambda\} \qquad (\lambda \text{ the empty word})$$

$$a \xmapsto{L} \{a\}$$

$$e + f \xmapsto{L} \text{union of } L(e) \text{ and } L(f)$$

$$e \cdot f \xmapsto{L} \text{element-wise concatenation of } L(e) \text{ and } L(f)$$

$$e^{*} \xmapsto{L} \text{set of “words over of } L(e)\text{”}$$

The Process Interpretation $\boldsymbol{P}$ (Milner)

$\mathbf{0} \quad \overset{\boldsymbol{P}}{\longmapsto} \quad$ deadlock δ

$\mathbf{1} \quad \overset{\boldsymbol{P}}{\longmapsto} \quad$ empty process ϵ

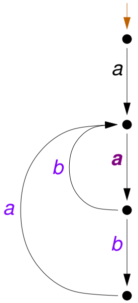
$a \quad \overset{\boldsymbol{P}}{\longmapsto} \quad$ atomic action a

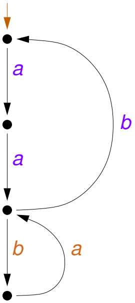
$e + f \quad \overset{\boldsymbol{P}}{\longmapsto} \quad$ alternative composition between $\boldsymbol{P}(e)$ and $\boldsymbol{P}(f)$

$e \cdot f \quad \overset{\boldsymbol{P}}{\longmapsto} \quad$ sequential composition of $\boldsymbol{P}(e)$ and $\boldsymbol{P}(f)$

$e^{*} \quad \overset{\boldsymbol{P}}{\longmapsto} \quad$ unbounded iteration of $\boldsymbol{P}(e)$

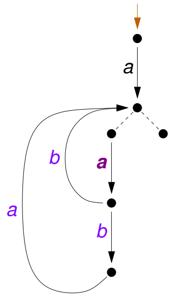
The Process Interpretation P

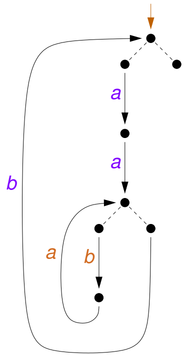




$$P\left(a\cdot\left(a\cdot(b+b\cdot a)\right)^{*}\cdot 0\right) \qquad P\left(\left(a\cdot a\cdot(b\cdot a)^{*}\cdot b\right)^{*}\cdot 0\right)$$

The Process Interpretation *P*





$$P\big(a\cdot(a\cdot(b+b\cdot a))^{*}\cdot 0\big) \qquad P\big((a\cdot a\cdot(b\cdot a)^{*}\cdot b)^{*}\cdot 0\big)$$

The Process Interpretation P (Transition System)

$$\frac{}{P(a) \xrightarrow{a} 1} \qquad \frac{}{1\downarrow}$$

$$\frac{P(e) \xrightarrow{a} P(e')}{P(e+f) \xrightarrow{a} P(e')} \qquad \frac{P(e)\downarrow}{P(e+f)\downarrow}$$

$$\frac{P(f) \xrightarrow{a} P(f')}{P(e+f) \xrightarrow{a} P(f')} \qquad \frac{P(f)\downarrow}{P(e+f)\downarrow} \qquad \frac{P(e)\downarrow \quad P(f)\downarrow}{P(e \cdot f)\downarrow}$$

$$\frac{P(e) \xrightarrow{a} P(e')}{P(e \cdot f) \xrightarrow{a} P(e' \cdot f)} \qquad \frac{P(e)\downarrow \quad P(f) \xrightarrow{a} P(f')}{P(e \cdot f) \xrightarrow{a} P(f')}$$

$$\frac{P(e) \xrightarrow{a} P(e')}{P(e^*) \xrightarrow{a} P(e' \cdot e^*)} \qquad \frac{}{P(e^*)\downarrow}$$

The Process Interpretation P (Transition System)

$$\frac{}{P(a) \xrightarrow{a} 1} \qquad \frac{}{1{\downarrow}}$$

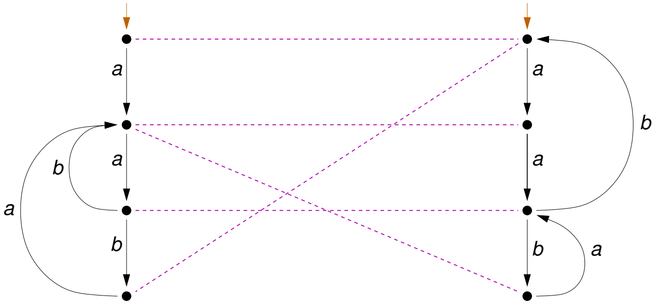
$$\frac{P(e) \xrightarrow{a} P(e')}{P(e+f) \xrightarrow{a} P(e')} \qquad \frac{P(e){\downarrow}}{P(e+f){\downarrow}}$$

$$\frac{P(f) \xrightarrow{a} P(f')}{P(e+f) \xrightarrow{a} P(f')} \qquad \frac{P(f){\downarrow}}{P(e+f){\downarrow}} \qquad \frac{P(e){\downarrow} \qquad P(f){\downarrow}}{P(e \cdot f){\downarrow}}$$

$$\frac{P(e) \xrightarrow{a} P(e')}{P(e \cdot f) \xrightarrow{a} P(e' \cdot f)} \qquad \frac{P(e){\downarrow} \qquad P(f) \xrightarrow{a} P(f')}{P(e \cdot f) \xrightarrow{a} P(f')}$$

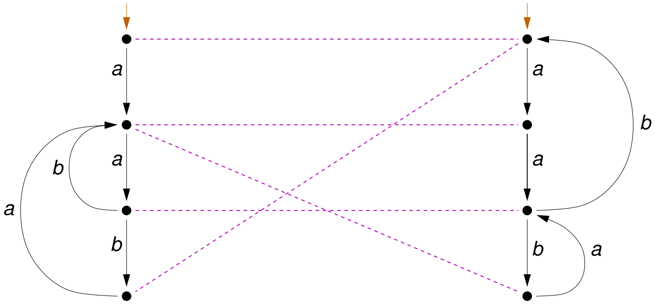
$$\frac{P(e) \xrightarrow{a} P(e')}{P(e^*) \xrightarrow{a} P(e' \cdot e^*)} \qquad \frac{}{P(e^*){\downarrow}}$$

Regular Expressions under Bisimulation



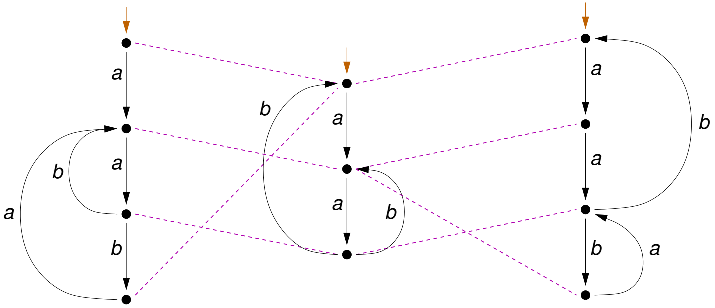
$$\boldsymbol{P}\big(a(a(b+ba))^*.0\big) \quad \Leftrightarrow \quad \boldsymbol{P}\big((aa(ba)^*a)^*.0\big)$$

Regular Expressions under Bisimulation



$$\big(a(a(b+ba))^*.0 \quad \underline{\leftrightarrow}_{P} \quad (aa(ba)^*a)^*.0$$

Regular Expressions under Bisimulation

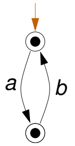


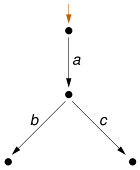
"*Two-exit iteration*"

$\notin im(\boldsymbol{P})$

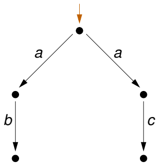
Properties of the Process Interpretation P

- There are finite transition graphs that are *not isomorpic* to any process graph $P(e)$ in the image of P.
- *What is more:* there are finite transition graphs that are *not bisimilar* to any process graph $P(e)$ in the image of P.
- Identities $e \leftrightarrow_P f$ under P also hold as identities $e =_L f$ under the language intepretation L. The converse is false:





$\not\leftrightarrow$



$$a \cdot (b + c) \quad \not\leftrightarrow_P \quad a \cdot b + a \cdot c$$

Milner's Questions (1984)

1. *Is a variant of Salomaa's axiomatization for language equivalence $=_L$ complete for $\leftrightarrow_P$?*

   – To my knowledge: Yet unsolved. (Partial & related results by Sewell; Fokkink; Corradini/De Nicola/Labella; G.)

2. *What structural property characterises the finite-state proc's that are bisimilar to proc's in the image of P ?*

   – Definiability by "well-behaved" specifications ([BC05]); this is decidable ([BCG05]).

3. *Does "minimal star height" over single-letter alphabets define a hierarchy modulo $\leftrightarrow_P$?*

   – Yes! (Hirshfeld and Moller, 1999).

Milner's Questions (1984)

1. *Is a variant of Salomaa's axiomatization for language equivalence $=_L$ complete for $\leftrightarrow_P$?*

   – To my knowledge: Yet unsolved. (Partial & related results by Sewell; Fokkink; Corradini/De Nicola/Labella; G.)

2. *What structural property characterises the finite-state proc's that are bisimilar to proc's in the image of P ?*

   – Definiability by "well-behaved" specifications ([BC05]); this is decidable ([BCG05]).

3. *Does "minimal star height" over single-letter alphabets define a hierarchy modulo $\leftrightarrow_P$?*

   – Yes! (Hirshfeld and Moller, 1999).

Milner's Questions (1984)

1. *Is a variant of Salomaa's axiomatization for language equivalence $=_L$ complete for $\leftrightarrow_P$?*
   – To my knowledge: Yet unsolved. (Partial & related results by Sewell; Fokkink; Corradini/De Nicola/Labella; G.)
2. *What structural property characterises the finite-state proc's that are bisimilar to proc's in the image of P ?*
   – Definiability by "well-behaved" specifications ([BC05]); this is decidable ([BCG05]).
3. *Does "minimal star height" over single-letter alphabets define a hierarchy modulo $\leftrightarrow_P$?*
   – Yes! (Hirshfeld and Moller, 1999).

Milner's Questions (1984)

1. *Is a variant of Salomaa's axiomatization for language equivalence $=_L$ complete for $\leftrightarrow_P$?*
   – To my knowledge: Yet unsolved. (Partial & related results by Sewell; Fokkink; Corradini/De Nicola/Labella; G.)
2. *What structural property characterises the finite-state proc's that are bisimilar to proc's in the image of **P** ?*
   – Definiability by "well-behaved" specifications ([BC05]); this is decidable ([BCG05]).
3. *Does "minimal star height" over single-letter alphabets define a hierarchy modulo $\leftrightarrow_P$?*
   – Yes! (Hirshfeld and Moller, 1999).

The Axiom System **REG** for $=_L$ (Salomaa's Axiomatization $\mathbf{F_1}$ reversed)

Axioms :

(B1)	$x + (y + z) = (x + y) + z$	(B7)	$x \cdot 1 = x$
(B2)	$(x \cdot y) \cdot z = x \cdot (y \cdot z)$	(B8)	$x \cdot 0 = 0$
(B3)	$x + y = y + x$	(B9)	$x + 0 = x$
(B4)	$(x + y) \cdot z = x \cdot z + y \cdot z$	(B10)	$x^* = 1 + x \cdot x^*$
(B5)	$x \cdot (y + z) = x \cdot y + x \cdot z$	(B11)	$x^* = (1 + x)^*$
(B6)	$x + x = x$		

Inference rules : equational logic *plus*

$$\frac{e = f \cdot e + g}{e = f^* \cdot g}\ \text{FIX}\ (\text{if } \lambda \notin \boldsymbol{L}(f))$$

Sound and Unsound Axioms of **REG** w.r.t. $\underline{\leftrightarrow}_P$

(B1) $x + (y + z) = (x + y) + z$

(B2) $(x \cdot y) \cdot z = x \cdot (y \cdot z)$

(B3) $x + y = y + x$

(B4) $(x + y) \cdot z = x \cdot z + y \cdot z$

(B5) $x \cdot (y + z) = x \cdot y + x \cdot z$

(B6) $x + x = x$

(B7) $x \cdot 1 = x$

(B8) $x \cdot 0 = 0$

(B9) $x + 0 = x$

(B10) $x^* = 1 + x \cdot x^*$

(B11) $x^* = (1 + x)^*$

Also sound are:

$$0 \cdot x = 0 \qquad \frac{e = f \cdot e + g}{e = f^* \cdot g}\ \text{FIX (if } \lambda \notin L(f))$$

Sound and Unsound Axioms of **REG** w.r.t. $\underline{\leftrightarrow}_P$

(B1)	$x + (y + z) = (x + y) + z$	(B7)	$x \cdot 1 = x$
(B2)	$(x \cdot y) \cdot z = x \cdot (y \cdot z)$	(B8)	$x \cdot 0 = 0$
(B3)	$x + y = y + x$	(B9)	$x + 0 = x$
(B4)	$(x + y) \cdot z = x \cdot z + y \cdot z$	(B10)	$x^* = 1 + x \cdot x^*$
(B5)	$x \cdot (y + z) = x \cdot y + x \cdot z$	(B11)	$x^* = (1 + x)^*$
(B6)	$x + x = x$		

Also sound are:

$$0 \cdot x = 0 \qquad \frac{e = f \cdot e + g}{e = f^* \cdot g} \text{ FIX (if } \lambda \notin L(f))$$

Milner's Adaptation for $\underline{\leftrightarrow}_P$: $\mathbf{BPA}^*_{0,1}$ +1-RSP

Axioms :

(B1)	$x + (y + z) = (x + y) + z$	(B7)	$x \cdot 1 = x$
(B2)	$(x \cdot y) \cdot z = x \cdot (y \cdot z)$	(B8)′	$0 \cdot x = 0$
(B3)	$x + y = y + x$	(B9)	$x + 0 = x$
(B4)	$(x + y) \cdot z = x \cdot z + y \cdot z$	(B10)	$x^* = 1 + x \cdot x^*$
		(B11)	$x^* = (1 + x)^*$
(B6)	$x + x = x$		

Inference rules : equational logic *plus*

$$\frac{e = f \cdot e + g}{e = f^* \cdot g}\ \text{1-RSP (if } \lambda \notin L(f))$$

Milner's Adaptation for $\underline{\leftrightarrow}_P$: $\mathbf{BPA}^*_{0,1}$ $+$1-RSP

Axioms :

(B1) $x + (y + z) = (x + y) + z$

(B2) $(x \cdot y) \cdot z = x \cdot (y \cdot z)$

(B3) $x + y = y + x$

(B4) $(x + y) \cdot z = x \cdot z + y \cdot z$

(B6) $x + x = x$

(B7) $x \cdot 1 = x$

(B8)′ $0 \cdot x = 0$

(B9) $x + 0 = x$

(B10) $x^* = 1 + x \cdot x^*$

(B11) $x^* = (1 + x)^*$

Inference rules : equational logic *plus*

$$\frac{e = f \cdot e + g}{e = f^* \cdot g}\ \text{1-RSP (if } \lambda \notin \boldsymbol{L}(f))$$

Overview

The Expressibility Problem

A finite-state process p is called
expressible as a regular expression under $\boldsymbol{P}$

iff

there exists $e \in RegExps$ such that $p \leftrightarrows \boldsymbol{P}(e)$.

The Expressibility Problem for $\boldsymbol{P}$
Instance: p a finite-state process
Question: Is p expressible as a regular expression under $\boldsymbol{P}$?

Well-Behaved Specifications (Motivation): A Correspondence Theorem

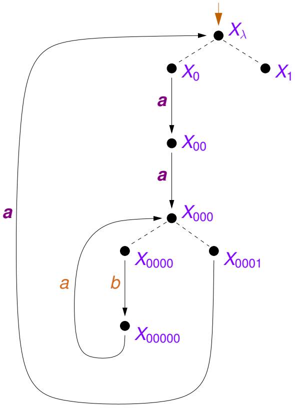
Theorem ([BC05])

Expressibility as a regular expression under ***P***
is equivalent to
definability by a well-behaved specification:

For all processes p ,

$$(\exists\, e \in \mathit{RegExps})\left[\, p \leftrightarrow \boldsymbol{P}(e) \,\right] \quad\Leftrightarrow\quad (\exists\, \mathcal{E} \in \mathit{WBSpecs})\left[\, p \text{ is a solution of } \mathcal{E} \,\right]$$

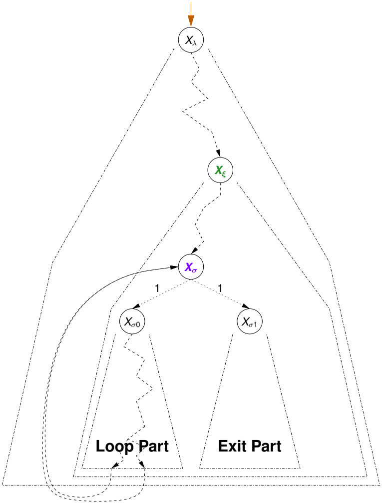
Well-Behaved Specifications (Example)



$$X_\lambda = 1 \cdot X_0 + 1 \cdot X_1$$
$$X_0 = \boldsymbol{a} \cdot X_{00}$$
$$X_{00} = \boldsymbol{a} \cdot X_{000}$$
$$X_{000} = 1 \cdot X_{0000} + 1 \cdot X_{0001}$$
$$X_{0000} = b \cdot X_{00000}$$
$$X_{00000} = a \cdot X_{000}$$
$$X_{0001} = \boldsymbol{a} \cdot X_\lambda$$
$$X_1 = 0$$

$$\boldsymbol{P}((\boldsymbol{aa}(ba)^{*}\boldsymbol{a})^{*}.0)$$

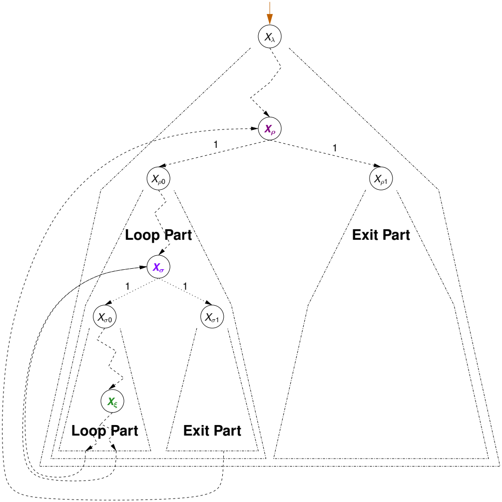
Well-Behaved Specifications (Some Intuition, I)



X_ξ, X_λ … *well-behaved variables*
(X_ξ “does not return” to a recursion variable above itself)

X_σ is a *cycling variable*
(Some recursion variable below X_σ “returns to” X_σ)

Well-Behaved Specifications (Some Intuition, II)



X_σ, X_ρ ... cycling variables

X_ξ *cycles back to* X_σ

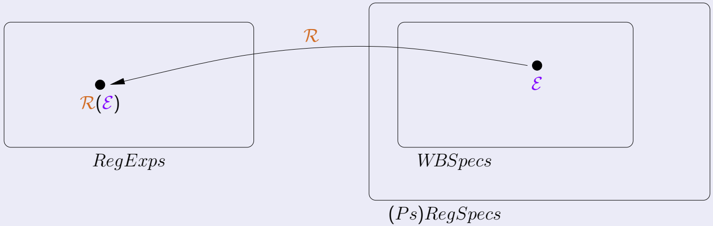
(The nearest return of X_ξ to a rec.var. above is to X_σ)

X_σ *cycles back to* X_ρ

Solvability Lemma

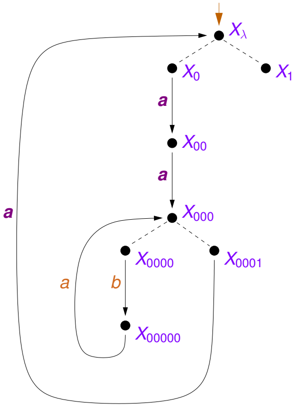
Lemma (Solvability of well-behaved spec's [BC05])

Every well-behaved specification is solved by a process represented by a regular expression.
Moreover: there is an effectively computable mapping $\mathcal{R} : \mathit{WBSpecs}(\Sigma) \to \mathit{RegExps}(\Sigma)$ *such that*



P$(\mathcal{R}(\mathcal{E}))$ *is a solution of* $\mathcal{E}$, *for all* $\mathcal{E} \in \mathit{WBSpecs}(\Sigma))$.

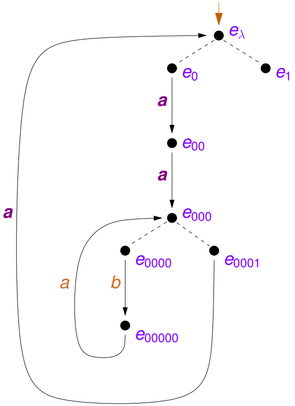
Well-Behaved Specifications (Example)



$P((aa(ba)^*a)^*.0)$

$$
\begin{aligned}
X_\lambda &= 1 \cdot X_0 + 1 \cdot X_1 \\
X_0 &= a \cdot X_{00} \\
X_{00} &= a \cdot X_{000} \\
X_{000} &= 1 \cdot X_{0000} + 1 \cdot X_{0001} \\
X_{0000} &= b \cdot X_{00000} \\
X_{00000} &= a \cdot X_{000} \\
X_{0001} &= a \cdot X_\lambda \\
X_1 &= 0
\end{aligned}
$$

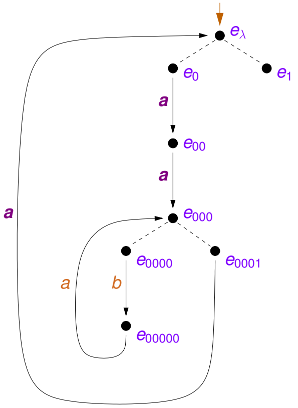
Solving a Well-Behaved Specification (Example, I/III)



$$P((aa(ba)^*a)^*.0)$$

$$
\begin{aligned}
e_\lambda &= 1 \cdot e_0 + 1 \cdot e_1 \\
e_0 &= a \cdot e_{00} \\
e_{00} &= a \cdot e_{000} \\
e_{000} &= 1 \cdot e_{0000} + 1 \cdot e_{0001} \\
e_{0000} &= b \cdot e_{00000} \\
e_{00000} &= a \cdot e_{000} \\
e_{0001} &= a \cdot e_\lambda \\
e_1 &= 0
\end{aligned}
$$

Solving a Well-Behaved Specification (Example, II/III)



$P((aa(ba)^*a)^*.0)$

$$e_{00000} = a \cdot e_{000}$$

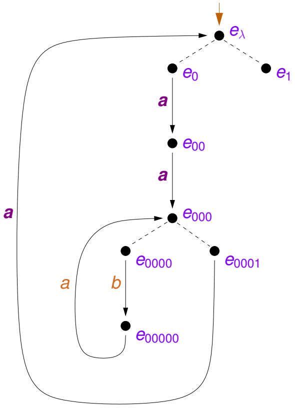
$$e_{0000} = b \cdot e_{00000} = b \cdot a \cdot e_{000}$$

$$e_{0001} = a \cdot e_\lambda$$

$$e_{000} = 1 \cdot e_{0000} + 1 \cdot e_{0001}$$
$$= b \cdot a \cdot e_{000} + a \cdot e_\lambda$$
$$\Rightarrow \quad e_{000} = (b \cdot a)^* \cdot a \cdot e_\lambda$$

(by 1-RSP)

Solving a Well-Behaved Specification (Example, III/III)



$P((aa(ba)^*a)^*.0)$

$$e_{000} = (b \cdot a)^* \cdot a \cdot e_\lambda$$

$$e_{00} = a \cdot e_{000} = a \cdot (b \cdot a)^* \cdot a \cdot e_\lambda$$

$$e_0 = a \cdot e_{00} = a \cdot a \cdot (b \cdot a)^* \cdot a \cdot e_\lambda$$

$$e_1 = 0$$

$$\begin{aligned} e_\lambda &= 1 \cdot e_0 + 1 \cdot e_1 \\ &= 1 \cdot a \cdot a \cdot (b \cdot a)^* \cdot a \cdot e_\lambda + 1 \cdot 0 \\ &= a \cdot a \cdot (b \cdot a)^* \cdot a \cdot e_\lambda + 0 \end{aligned}$$

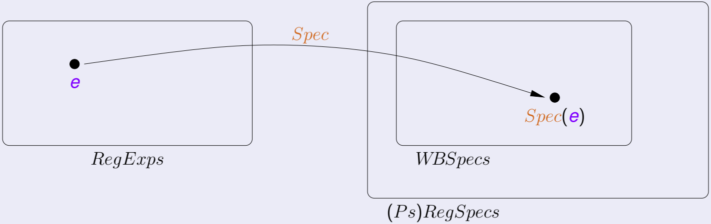
$$\Rightarrow \; e_\lambda = (a \cdot a \cdot (b \cdot a)^* \cdot a)^* \cdot 0$$

(by 1-RSP)

Definability Lemma

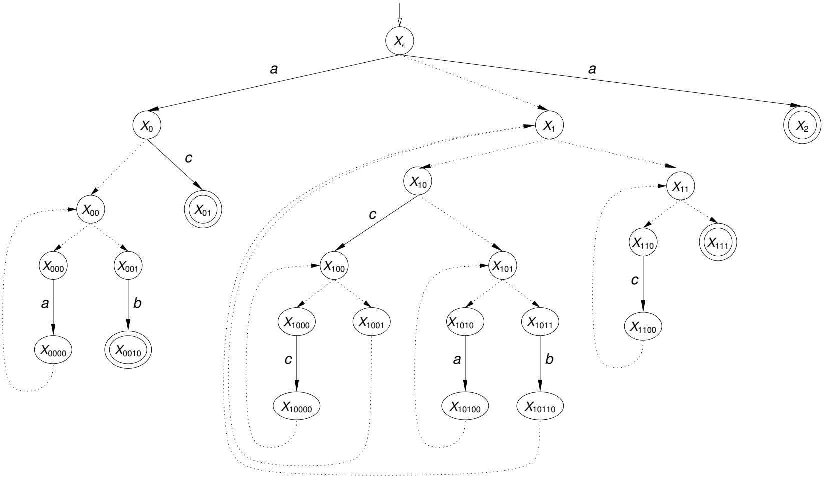
Lemma (Definability by well-behaved spec's [BC05])

*The processes represented by regular expressions under **P** are definable by well-behaved specifications.*
Moreover: there is an effectively computable mapping
$Spec : RegExps(\Sigma) \to WBSpecs(\Sigma)$ *such that*

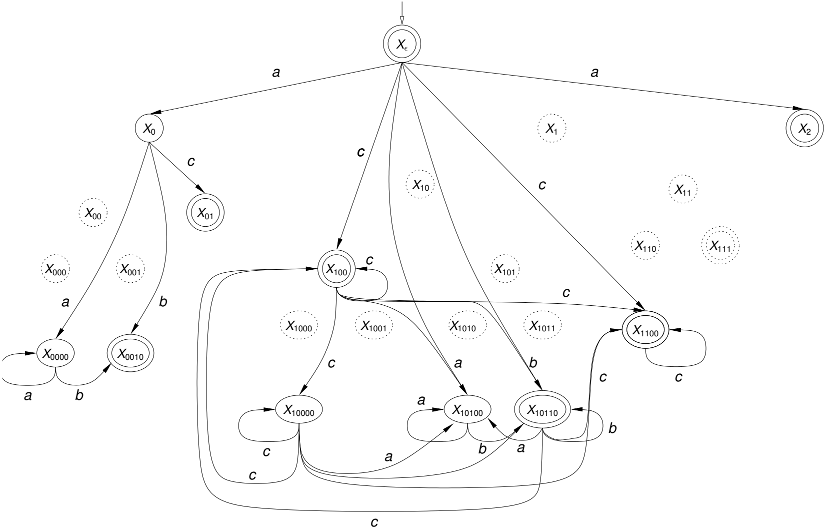


for all $e \in RegExps(\Sigma)$, **P**(e) *is a solution of* $Spec(e)$.

Example: $Spec(a(a^*b + c) + (c^* + a^*b)^* + a)$



Example: Canonical Solution of *Spec*$(a(a^*b + c) + \ldots)$



The Correspondence Theorem

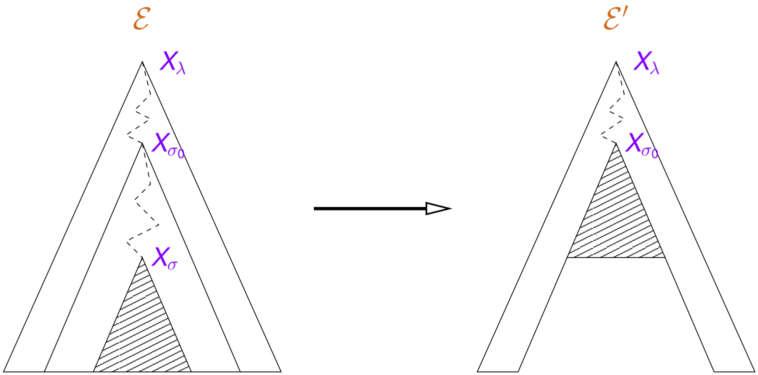
Theorem ([BC05])

Expressibility as a regular expression under ***P***
is equivalent to
definability by a well-behaved specification:

For all processes p ,

$$(\exists\, e \in \mathit{RegExps})\,\big[\, p \leftrightarrows \boldsymbol{P}(e) \,\big] \;\Leftrightarrow\; (\exists\, \mathcal{E} \in \mathit{WBSpecs})\,\big[\, p \text{ is a solution of } \mathcal{E} \,\big]$$

Reducible Well-Behaved Specifications (Example)



$\langle X_\sigma \,|\, \mathcal{E} \rangle \Leftrightarrow \langle X_{\sigma_0} \,|\, \mathcal{E} \rangle$

X_σ, X_{σ_0} are well-behaved

Reducibility Lemma, Decidability Theorem

Lemma (Reducibility of well-behaved spec's [BCG05])

Let $\mathcal{E}$ be a well-behaved specification that has a finite-state process p with n states and maximal branching degree k as a solution.

Then $\mathcal{E}$ is equivalent to a well-behaved specification $\mathcal{E}_{\boldsymbol{red}}$ with

- *depth less or equal to $(n+1)^3 \cdot 2^{3k}$, and*
- *less or equal to k summands in each defining equation.*

Theorem ([BCG05])

*Expressibility by a regular expression under the process interpretation is decidable. In other words, the expressibility problem under **P** is algorithmically solvable.*

Overview

The Star Height Problem

The Star Height Problem for *P*

Instance: $e \in RegExps(\Sigma)$

Question: What is the minimal star height of e under P?

Milner's Star-Height Question

Does "minimal star height" modulo $\leftrightarrows_P$ define a hierarchy also over single-letter alphabets?

Star Height, and Star Height of Regular Languages

The *star height* $sh(e)$ of a regular expression e is the maximum number of nested stars in e.

For example: $sh(\,(a+b)c\,) = 0\,,\;\; sh(\,(a(ba)^{*}a)^{*}dc^{*}\,) = 2\,.$

Definition

The *(restricted) star height* $sh(L)$ of a regular language L is the least natural number n such that $sh(e) = n$ for some regular expression e that represents L.

Generalised Star Height: concerning
generalised regular expressions in which
complementation and *intersection* may occur.

Classical Results on (Restricted) Star Height

1. *Every regular language over a single-letter alphabet has star height 1 at most.*

2. *There are regular languages with any preassigned star height* (Eggan, 1963);
   ... even over a two-letter alphabet (McNaughton, 1965, Dejean/Schützenberger, 1966);

3. *There exists an algorithm for computing the star height of the regular language given by a regular expression* (Hashiguchi, 1983).
   (The (Restricted) Star Height Problem is solvable).

Minimal Star Height under *P*

Definition

The *minimal star height* $msh(e)$ *(under **P**)* of a regular expression e is the least natural number n such that there exists a regular expression e_{min} with $sh(e_{min}) = n$ and $e_{min} \leftrightarrows_{P} e$.

Remark. For all $e \in RegExps$ it holds: $sh(L(e)) \leq msh(e)$.

Results for Minimal Star Height under *P*

1. *For every $n \in \mathbb{N}$, there exists a regular expression f_n over the single-letter alphabet such that the minimal star height of f_n is n* (Hirshfeld/Moller, 2000).
2. Consequently: *For the set regular expressions over a non-empty alphabet, "minimal star height under **P**" defines a proper hierarchy.*
3. The Star-Height Problem under ***P*** is solvable ([BCG05]).

The Star Height Problem under *P*
Instance: $e \in RegExps(\Sigma)$
Question: What is the minimal star height of e under ***P***?

Overview

The Axiomatization Problem(s)

1. Is Milner's adaptation $\mathbf{BPA}^{*}_{\mathbf{0},\mathbf{1}} + \text{1-RSP}$ of Salomaa's complete axiomatization $\mathbf{F_1}$ for $=_L$ complete for $\leftrightarrows_P$?

   Is there a finite extension of $\mathbf{BPA}^{*}_{\mathbf{0},\mathbf{1}} + \text{1-RSP}$ (by additional axioms or rules) that is complete for $\leftrightarrows_P$?

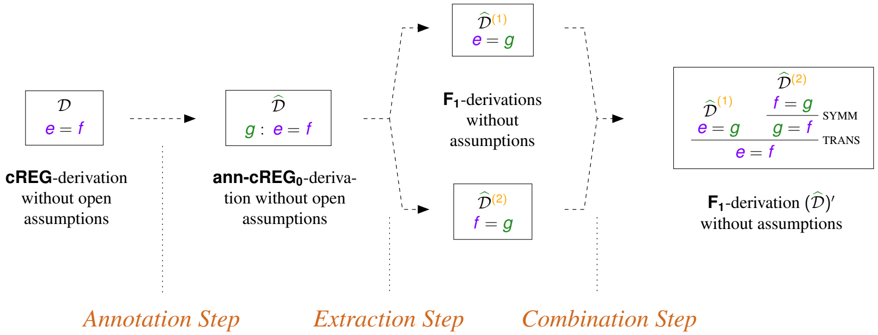
2. Is there a natural-deduction style or sequent-style proof system that is complete for $\leftrightarrows_P$?

Inspiration: A Coinductive/Proof-Theoretic Completeness Proof

In [G05] a coinductive/proof-theoretic proof is given for the completeness of Salomaa's axiomatisation $\mathbf{F_1}$ w.r.t. $=_L$:

1. A characterisation of $=_L$ by a "finitary coinduction principle" (based on "Brzozowski derivatives"):
$$e =_L f \iff e \sim_{\text{fin}} f\,.$$
2. A natural-deduction system **cREG** that is sound and complete with respect to $=_L$ (reminiscent of a system by Brandt/Henglein, 1998).
3. A proof-transformation from **cREG** to Salomaa's complete axiomatisation $\mathbf{F_1}$ of $=_L$.

The Proof-Transformation from **cREG** to $\mathbf{F_1}$



A Coinductive/Proof-Theoretic Completeness Proof for an Extension of Milner's system

Here we describe a similar coinductive/proof-theoretic completeness proof w.r.t. $\leftrightarrows_P$ for an extension of $\mathbf{BPA}^*_{0,1}+$1-RSP by a more powerful rule USP:

1. A characterisation of $\leftrightarrows_P$ by a "finitary coinduction principle" (based on "Antimirov derivatives"):
$$e \leftrightarrows_P f \quad \Longleftrightarrow \quad e \sim_{\text{fin}} f\,.$$
2. A natural-deduction system $\mathbf{c\text{-}BPA}^*_{0,1}$ that is sound and complete with respect to $\leftrightarrows_P$.
3. A proof-transformation from $\mathbf{c\text{-}BPA}^*_{0,1}$ to an extension $\mathbf{BPA}^*_{0,1}+$USP of Milner's system $\mathbf{BPA}^*_{0,1}+$1-RSP.

Antimirov and Brzozowksi Derivatives

Brzozowski deriv's (1963) Antimirov's partial derivatives (1995)

$$(\cdot)_{\cdot} : \mathcal{R}(\Sigma) \times \Sigma \to \mathcal{R}(\Sigma) \qquad \partial : \mathcal{R}(\Sigma) \times \Sigma \to \mathcal{P}_{\mathsf{f}}(\mathcal{R}(\Sigma))$$

$$\langle e, a \rangle \mapsto e_a \qquad \langle e, a \rangle \mapsto \partial_a(e)$$

- Brzozowski der's mimic language derivatives on a synatactic level: $\boldsymbol{L}(e_a) = (\boldsymbol{L}(e))_a \; (=_{\text{def}} \{ v \mid a.v \in \boldsymbol{L}(e) \})$.
- Partial der's are mathematically motivated refinements.
- Both defined syntactically by ind. on the size of reg. expr's.
- Relationship: F.a. $e \in RegExps(\Sigma)$, $e_a \equiv_{\mathbf{ACI}} \sum_{e' \in \partial_a(e)} e'$
- *Every regular expression has only finitely many Brzozowski (Antimirov) derivatives.*

The Coalgebra Induced by Partial Derivatives

Antimirov's partial derivatives induce an F-coalgebra $(\mathit{RegExps}(\Sigma), \langle o, t\rangle)$, for the functor $F(X) = 2 \times \mathcal{P}_{\mathrm{f}}(\Sigma \times X)$ by:

$$\langle o, t\rangle : \mathit{RegExps}(\Sigma) \longmapsto 2 \times \mathcal{P}_{\mathrm{f}}(\Sigma \times \mathit{RegExps}(\Sigma)), \text{ where}$$

$$o : \mathit{RegExps}(\Sigma) \longrightarrow 2$$
$$e \longmapsto o(e) =_{\mathrm{def}} \begin{cases} 0 & \ldots \boldsymbol{P}(e)\not\downarrow \;\; (\lambda \notin \boldsymbol{L}(e)) \\ 1 & \ldots \boldsymbol{P}(e)\downarrow \;\; (\lambda \in \boldsymbol{L}(e)) \end{cases}$$

$$t : \mathit{RegExps}(\Sigma) \longrightarrow \mathcal{P}_{\mathrm{f}}(\Sigma \times \mathit{RegExps}(\Sigma))$$
$$e \longmapsto t(e) =_{\mathrm{def}} \{\langle a, e'\rangle \mid a \in \Sigma,\ e' \in \partial_a e\}\ .$$

$\sim$: bisimilarity on this coalgebra;

$e \sim_{\mathrm{fin}} f$: there is a finite bisimulation between e and f.

Relationship with the Process Interpretation *P*

Lemma

For all $e, f \in RegExps(\Sigma)$ *and* $a \in \Sigma$ *:*

$$\left[\, P(e) \xrightarrow{a} P(f) \iff f \in \partial_a(e) \,\right].$$

A *finitary coinduction principle* (*finite bisimulation principle*):

Theorem

For all $e, f \in RegExps(\Sigma)$ *:*

$$e \leftrightarrows_{P} f \iff e \sim_{fin} f \text{ in } (RegExps(\Sigma), \langle o, t \rangle).$$

The Proof System $\mathbf{c\text{-}BPA}^*_{0,1}$

Inference rule in $\mathbf{c\text{-}BPA}^*_{0,1}$: (Given $\Sigma = \{a_1, \ldots, a_n\}$).

$$\frac{\ldots \quad \begin{array}{c}[e = f]^{\boldsymbol{u}} \\ \mathcal{D}_1^{(i)} \\ e_1^{(i)} = f_1^{(i)}\end{array} \quad \ldots \quad \begin{array}{c}[e = f]^{\boldsymbol{u}} \\ \mathcal{D}_{m_i}^{(i)} \\ e_{m_i}^{(i)} = f_{m_i}^{(i)}\end{array} \quad \ldots}{e = f} \;\text{c-COMP}, \boldsymbol{u} \text{ (if (*))}$$

where (*) demands:

- $o(e) = o(f)$ holds, and
- $\partial_{a_i} e = \{e_1^{(i)}, \ldots, e_{m_i}^{(i)}\}$ and $\partial_{a_i} f = \{f_1^{(i)}, \ldots, f_{m_i}^{(i)}\}$
  (for all $i \in \{1, \ldots, n\}$).

A Derivation in $\mathbf{c\text{-}BPA}^*_{0,1}$

For $e =_{\text{def}} 1\cdot(a\cdot a\cdot(b\cdot a)^*\cdot b)^*\cdot 0$ and $f =_{\text{def}} a\cdot(a\cdot(b+b\cdot a)^*)\cdot 0$, for which $e \leftrightarrows_P f$ holds, we find the following proof in $\mathbf{c\text{-}BPA}^*_{0,1}$:

$$\dfrac{\dfrac{\dfrac{\dfrac{(e_1 = f_1)^{u}}{e = f_3}\,\text{COMP} \qquad \dfrac{(e_2 = f_2)^{v}}{e_3 = f_1}\,\text{COMP}}{e_2 = f_2}\,\text{c-COMP},\, v}{e_1 = f_1}\,\text{c-COMP},\, u}{e = f}\,\text{COMP}$$

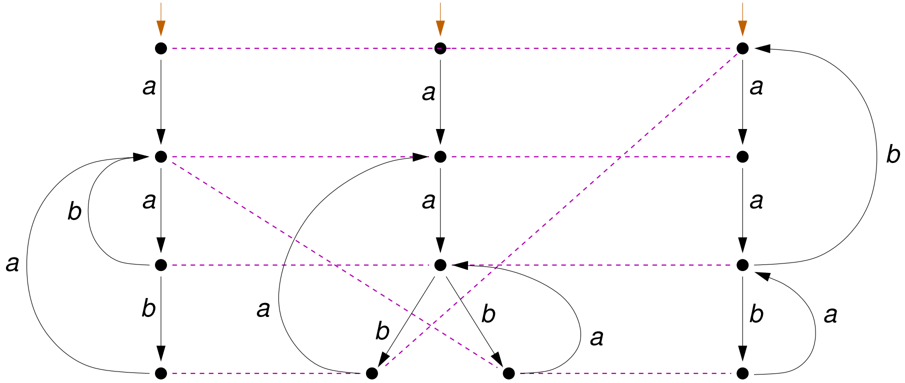
where, in particular,

$$e_2 \equiv 1\cdot(b\cdot a)^*\cdot b\cdot(a\cdot a\cdot(b\cdot a)^*\cdot b)^*\cdot 0\,,$$

$$f_2 \equiv 1\cdot(b + b\cdot a)\cdot(a\cdot(b + b\cdot a))^*\cdot 0\,,$$

$$\partial_b(e_2) = \{e, e_3\} \text{ and } \partial_b(f_2) = \{f_1, f_3\}\,.$$

A Derivation in $\mathbf{c\text{-}BPA}^*_{0,1}$ (the Intuition)



Completeness of $\mathbf{c\text{-}BPA}^*_{0,1}$

Theorem

$\mathbf{c\text{-}BPA}^*_{0,1}$ *is sound and complete w.r.t.* $\leftrightarrows_P$:

$$(\forall e, f \in RegExps(\Sigma)) \left[\vdash_{\mathbf{c\text{-}BPA}^*_{0,1}} e = f \iff e \leftrightarrows_P f \right].$$

Proof.

By the finitary coinduction principle for $\leftrightarrows_P$. □

Reconstructing Regular Expressions from Partial Derivatives

Lemma

Let $\Sigma = \{a_1, \ldots, a_n\}$ *. Then for all* $e \in RegExps(\Sigma)$ *it holds:*

$$\vdash_{\mathbf{BPA}^*_{0,1}} e = o(e) + \sum_{i=1}^{n} \sum_{e' \in \partial_{a_i}(e)} a_i . e' \ .$$

(This statement is reminiscent of the *fundamental theorem of calculus* that links differentiation and integration.)

Unique Solvability Principle(s)

$$\text{1-RSP}\ \frac{x = f.x + g}{x = f^{*}.g}\ \ (\text{if } \lambda \notin \boldsymbol{L}(f))$$

$$\text{1-USP}\ \frac{x = f.x + g \qquad y = f.y + g}{x = y}\ \ (\text{if } \lambda \notin \boldsymbol{L}(f))$$

$$\text{USP}\ \frac{\{\ x_j = E_j(x_1, \ldots, x_m)\ \}_{j=1}^{m} \qquad \{\ y_j = E_j(y_1, \ldots, y_m)\ \}_{j=1}^{m}}{x_i = y_i}$$

where, for all $i \in \{1, \ldots, m\}$,

$E_j(x_1, \ldots, x_m)$ is of the form $[1+] \sum_{k=1}^{m_j} a_{l_k}.x_{l_{j,k}}$.

Transforming into $\mathbf{BPA}^*_{\mathbf{0},\mathbf{1}}+$ USP-der's (Example)

By the "expr's reconstr. lemma", one finds that in the example the vectors $\langle e, e_1, e_2, e, e_3\rangle$ and $\langle f, f_1, f_2, f_3, f_1\rangle$ of reg. expr's satisfy the same system of equations. This enables to extract from the proof in $\mathbf{c\text{-}BPA}^*_{\mathbf{0},\mathbf{1}}$ a proof in $\mathbf{BPA}^*_{\mathbf{0},\mathbf{1}}+$USP:

$$e \overset{\vdots}{=} a.e_1 \qquad e_3 \overset{\vdots}{=} a.e_2 \qquad\qquad f_3 \overset{\vdots}{=} a.f_1 \qquad f_1 \overset{\vdots}{=} a.f_2$$

$$e_2 \overset{\vdots}{=} b.e + b.e_3 \qquad\qquad f_2 \overset{\vdots}{=} b.f_3 + b.f_1$$

$$e_1 \overset{\vdots}{=} a.e_2 \qquad\qquad f_1 \overset{\vdots}{=} a.f_2$$

$$\frac{e \overset{\vdots}{=} a.e_1 \qquad\qquad f \overset{\vdots}{=} a.f_1}{e = f}\ \text{USP}$$

Completeness of $\mathbf{BPA}^{*}_{0,1}$+USP

Theorem

$\mathbf{BPA}^{*}_{0,1}$+USP *is sound and complete w.r.t.* $\leftrightarrows_{P}$:

$$(\forall e, f \in RegExps)\ \big[\ \vdash_{\mathbf{BPA}^{*}_{0,1}+\mathrm{USP}} e = f \iff e \leftrightarrows_{P} f\ \big]\ .$$

Remaining Question (equivalent to Milner's first question):

Is $\mathbf{BPA}^{*}_{0,1}$+1-USP *complete for* $\leftrightarrows_{P}$ *?*

Local Summary

- Antimirov's partial derivatives guide the operational behaviour of regular expressions under ***P***.
- The complete proof system $\mathbf{c\text{-}BPA}^*_{0,1}$ for $\Leftrightarrow_P$ which is based on a "finitary coinduction principle" for $\Leftrightarrow_P$.
- Replacing 1-RSP in Milner's system $\mathbf{BPA}^*_{0,1}$+1-RSP by the *unique solvability principle* USP gives the complete axiomatization $\mathbf{BPA}^*_{0,1}$+USP for $\Leftrightarrow_P$.

Global Summary

- The **Expressibility Problem** for ***P*** is solvable.
- The **Star-Height Problem** for ***P*** is solvable.
- Concerning the **Axiomatisation Problem** for $\Leftrightarrow_{\boldsymbol{P}}$:
  - There is a coinductively motivated, natural-deduction system $\mathbf{c\text{-}BPA}^{*}_{\mathbf{0},\mathbf{1}}$ that is complete for $\Leftrightarrow_{\boldsymbol{P}}$.
  - The system $\mathbf{BPA}^{*}_{\mathbf{0},\mathbf{1}} + \mathrm{USP}$ is complete for $\Leftrightarrow_{\boldsymbol{P}}$ (USP is a *unique solvability principle* for linear systems of equations).
  - Milner's question: "Is $\mathbf{BPA}^{*}_{\mathbf{0},\mathbf{1}} + \text{1-RSP}$ complete for $\Leftrightarrow_{\boldsymbol{P}}$?" is (to my knowledge) still unanswered.

A Further Partial Result. Questions.

Let $\rightarrow_P$ denote the relation functional bisimulation on well-behaved specifications, and $\leftarrow_P$ its converse.

Theorem

Let $e, f \in RegExps(\Sigma)$. Then it holds:

$$Spec(e)\ (\rightarrow_P \cup \leftarrow_P)^*\ Spec(f) \implies \vdash_{\mathbf{BPA}^*_{0,1}+\text{1-RSP}} e = f \qquad (1)$$

Questions:

1. Does the converse of (1) hold? (My Conjecture is: No)
2. What relation on corresponding well-behaved spec's does provability in $\mathbf{BPA}^*_{0,1}$+1-RSP induce?
   (Having a grip on this relation could help to prove/disprove completeness of $\mathbf{BPA}^*_{0,1}$+1-RSP w.r.t. $\leftrightarrow_P$.)

A Further Partial Result. Questions.

Let $\rightarrow_P$ denote the relation functional bisimulation on well-behaved specifications, and $\leftarrow_P$ its converse.

Theorem

Let $e, f \in RegExps(\Sigma)$. Then it holds:

$$Spec(e)\ (\rightarrow_P \cup \leftarrow_P)^*\ Spec(f) \implies \vdash_{\mathbf{BPA}^*_{0,1}+\text{1-RSP}} e = f \qquad (1)$$

Questions:

1. Does the converse of (1) hold? (My Conjecture is: No)
2. What relation on corresponding well-behaved spec's does provability in $\mathbf{BPA}^*_{0,1}+$1-RSP induce?
(Having a grip on this relation could help to prove/disprove completeness of $\mathbf{BPA}^*_{0,1}+$1-RSP w.r.t. $\leftrightarrow_P$.)

A Further Partial Result. Questions.

Let $\rightharpoondown_P$ denote the relation functional bisimulation on well-behaved specifications, and $\leftharpoondown_P$ its converse.

Theorem

Let $e, f \in RegExps(\Sigma)$. *Then it holds:*

$$Spec(e)\ (\rightharpoondown_P \cup \leftharpoondown_P)^*\ Spec(f) \implies \vdash_{\mathbf{BPA}^*_{0,1}+\text{1-RSP}} e = f \qquad (1)$$

Questions:

1. Does the converse of (1) hold? (My Conjecture is: No)
2. What relation on corresponding well-behaved spec's does provability in $\mathbf{BPA}^*_{0,1}$+1-RSP induce?
   (Having a grip on this relation could help to prove/disprove completeness of $\mathbf{BPA}^*_{0,1}$+1-RSP w.r.t. $\leftrightarrow_P$.)

A Further Partial Result. Questions.

Let $\rightarrow_P$ denote the relation functional bisimulation on well-behaved specifications, and $\leftarrow_P$ its converse.

Theorem

Let $e, f \in RegExps(\Sigma)$ *. Then it holds:*

$$Spec(e)\ (\rightarrow_P \cup \leftarrow_P)^*\ Spec(f) \implies \vdash_{\mathbf{BPA}^*_{0,1}+\text{1-RSP}} e = f \qquad (1)$$

Questions:

1. Does the converse of (1) hold? (My Conjecture is: No)
2. What relation on corresponding well-behaved spec's does provability in $\mathbf{BPA}^*_{0,1}$+1-RSP induce?
(Having a grip on this relation could help to prove/disprove completeness of $\mathbf{BPA}^*_{0,1}$+1-RSP w.r.t. $\leftrightarrow_P$.)

References I

V. Antimirov
Partial derivatives of regular expressions and finite automaton constructions
Theoretical Computer Science, 1996.

J.C.M. Baeten and F. Corradini
Regular Expressions in Process Algebra.
Proceedings of LICS 2005, 2005.

J.C.M. Baeten, F. Corradini, and C. Grabmayer
A Characterization of Regular Expressions under Bisimulation.
Research Report, Techn. Univ. Eindhoven, Oct. 2005.

References II

- C. Grabmayer
  Using Proofs by Coinduction to Find "Traditional" Proofs
  Proceedings of CALCO 2005, LNCS 3629, 2005.

- R. Milner
  A Complete Inference System for a Class of Regular Behaviours.
  Journal of Comp. and Syst. Sciences, 28:439-466, 1984.

- Y. Hirshfeld and F. Moller
  On the Star Height of Unary Regular Behaviours.
  In: *Proof, Language, Interaction: Essays in honour of Robin Milner*, MIT Press, 2000.