Contents

1. Equivalence of reductions

Problem 1 *When are two reductions equivalence?*

Solution 2 *If they perform the same steps, possibly in a different order.*

1.1. Running example

TRS with rules

$$\begin{aligned} a &\to b \\ f(x,b) &\to g(x) \\ g(b) &\to c \end{aligned}$$

Two reductions from $f(a,a)$ to the term c.

$$\mathcal{R} \ : \ f(\tilde{a},a) \to f(b,\overline{a}) \to \underline{f}(b,\underline{b}) \to \underline{g}(\underline{b}) \to c$$

$$\mathcal{S} \ : \ f(a,\overline{a}) \to \underline{f}(a,\underline{b}) \to g(\tilde{a}) \to \underline{g}(\underline{b}) \to c$$

Reductions intuitively equivalent

Home Page
Title Page
Contents
Page 6 of 56
Go Back
Full Screen
Close
Quit

2. Distinguishing steps

2.1. Syntactic accidents

Ex 3 *TRS* $\mathcal{T}$ *has rule* $\varrho : f(x) \to x$.
Two steps from $f(f(a))$*:*

$$\underline{f}(f(a)) \to f(a) \qquad f(\underline{f}(a)) \to f(a)$$

Both steps give rise to the step $f(f(a)) \to_{\mathcal{T}} f(a)$ *in the underlying abstract rewriting system* $\to_{\mathcal{T}}$ *of* $\mathcal{T}$*: a syntactic accident.*

Problem 4 *Steps have no identity*

2. Distinguishing steps

2.1. Syntactic accidents

Ex 3 *TRS $\mathcal{T}$ has rule $\varrho : f(x) \to x$.*
Two steps from $f(f(a))$:

$$\underline{f}(f(a)) \to f(a) \qquad f(\underline{f}(a)) \to f(a)$$

Both steps give rise to the step $f(f(a)) \to_{\mathcal{T}} f(a)$ in the underlying abstract rewriting system $\to_{\mathcal{T}}$ of $\mathcal{T}$: a syntactic accident.

Problem 4 *Steps have no identity*

Solution 5 *Provide steps with identity*

2.2. Abstract rewriting systems

Def 6 *An abstract rewriting system, ARS for short, is a quadruple* $\langle A, \Phi, \mathsf{src}, \mathsf{tgt} \rangle$ *with*

1. A *set of objects*
2. Φ *set of steps*
3. $\mathsf{src}, \mathsf{tgt} : \Phi \to A$ *the source and target functions*

$$\text{object} \xleftarrow{\ \mathsf{src}\ } \text{step} \xrightarrow{\ \mathsf{tgt}\ } \text{object}$$

– $\to, \rightsquigarrow, \Rightarrow, \ldots$ range over ARSs

– $a, b, c, \ldots$ range over objects

– $\phi, \psi, \chi, \ldots$ range over steps

Home Page

Title Page

Contents

◀◀ ▶▶

◀ ▶

Page 8 of 56

Go Back

Full Screen

Close

Quit

2.3. Examples of ARSs

Exs 7 1. *The black hole ARS* $\circlearrowright$ *has a single object* $\bullet$ *and a single step from the object to itself.*

2. *For every natural number* n, *the ARS* $\rightarrow^n$ *has objects* $\bullet_1, \ldots, \bullet_n$ *and steps* $i+1 : \bullet_i \rightarrow^n \bullet_{i+1}$, *for every* i *such that* $1 \leq i$ *and* $i+1 \leq n$.

   Special cases: the empty ARS $\rightarrow^0$ *and the single-object ARS* $\rightarrow^1$

3. *The ARS* $\multimap\!\!\infty\!\!\rightarrow$ *is union of all* $\rightarrow^n$: *the infinite straight line* $\bullet_1 \rightarrow^1 \bullet_2 \rightarrow^2 \bullet_3 \rightarrow^3 \cdots$.

4. *The syntactic accident ARS* $\rightrightarrows$ *consists of two objects and two steps, as indicated.*

5. *The diamond ARS* $\Diamond$ *is defined in the obvious way*

2.4. Composites

Def 8 *The reflexive–transitive closure* $\longrightarrow^*$ *of an abstract rewriting system* $\longrightarrow = \langle A, \Phi, \mathsf{src}, \mathsf{tgt}\rangle$ *is:*

– A *is the set of objects.*

– *Steps with source, target:*

$$\frac{a \in A}{1_a : a \longrightarrow^* a} \quad \frac{\phi : a \longrightarrow b \in \Phi}{\phi : a \longrightarrow^* b} \quad \frac{\phi : a \longrightarrow^* b \quad \psi : b \longrightarrow^* c}{(\phi \cdot \psi) : a \longrightarrow^* c}$$

– *The empty step for an object* a *is* $1_a : a \longrightarrow^* a$

– $(\phi \cdot \psi)$ *is a composite step*

$(\phi \cdot \psi) \cdot \chi$ and $\phi \cdot (\psi \cdot \chi)$ are (distinct) steps of $\longrightarrow^*$

2.5. Reductions

Def 9 $\twoheadrightarrow$ *is* $\longrightarrow^*$ *modulo reduction identities*

$$
\begin{array}{rcl}
1 \cdot \phi & \approx & \phi \\
\phi \cdot 1 & \approx & \phi \\
(\phi \cdot \psi) \cdot \chi & \approx & \phi \cdot (\psi \cdot \chi)
\end{array}
$$

Steps, $\mathcal{R}$, $\mathcal{S}$, $\mathcal{P}$, *of* $\twoheadrightarrow$ *are called (finite) reductions.*

Problem 10 *How to choose unique representatives?*

2.5. Reductions

Def 9 $\twoheadrightarrow$ *is* $\longrightarrow^*$ *modulo reduction identities*

$$\begin{array}{rcl} 1 \cdot \phi & \approx & \phi \\ \phi \cdot 1 & \approx & \phi \\ (\phi \cdot \psi) \cdot \chi & \approx & \phi \cdot (\psi \cdot \chi) \end{array}$$

Steps, $\mathcal{R}$, $\mathcal{S}$, $\mathcal{P}$, *of* $\twoheadrightarrow$ *are called (finite) reductions.*

Problem 10 *How to choose unique representatives?*

Solution 11 *Orient equations into complete TRS:*

$$\begin{array}{rcl} 1 \cdot x & \Rightarrow & x \\ x \cdot 1 & \Rightarrow & x \\ (x \cdot y) \cdot z & \Rightarrow & x \cdot (y \cdot z) \end{array}$$

Home Page
Title Page
Contents
Page 11 of 56
Go Back
Full Screen
Close
Quit

2.6. Term rewriting systems

Def 12 *A* *term rewriting system*, *TRS for short,* $\mathcal{T}$ *is a structure* $\langle \Sigma, R \rangle$ *such that*

- Σ *is an alphabet*
- R *is an abstract rewriting system having terms over* Σ *(and variables) as objects*

$$\text{term} \xleftarrow{\text{lhs}} \text{rule} \xrightarrow{\text{rhs}} \text{term}$$

$\varrho, \vartheta, \varsigma, \ldots$ range over rules

Problem 13 *How do generated steps look like?*

Home Page
Title Page
Contents
◀◀ ▶▶
◀ ▶
Page 11 of 56
Go Back
Full Screen
Close
Quit

2.6. Term rewriting systems

Def 12 *A term rewriting system, TRS for short,* $\mathcal{T}$ *is a structure* $\langle \Sigma, R \rangle$ *such that*

- Σ *is an alphabet*
- R *is an abstract rewriting system having terms over* Σ *(and variables) as objects*

$$\text{term} \xleftarrow{\text{lhs}} \text{rule} \xrightarrow{\text{rhs}} \text{term}$$

$\varrho, \vartheta, \varsigma, \ldots$ range over rules

Problem 13 *How do generated steps look like?*

Solution 14 *Proof terms*

2.7. Proof terms

Def 15 *The proof term alphabet is the (disjoint) union of Σ, the set of rule symbols of R, and $\{\cdot\}$, where $\cdot$ is the binary composition symbol. The underlying ARS $\geq_{\mathcal{T}}$*

- *Objects are terms over Σ.*
- *Steps are proof terms defined by*

$$\frac{\phi_1 : s_1 \geq t_1 \quad \dots \quad \phi_n : s_n \geq t_n}{f(\phi_1, \dots, \phi_n) : f(s_1, \dots, s_n) \geq f(t_1, \dots, t_n)} \text{ (repl)}$$

$$\frac{\phi_1 : s_1 \geq t_1 \quad \dots \quad \phi_n : s_n \geq t_n}{\varrho(\phi_1, \dots, \phi_n) : l(s_1, \dots, s_n) \geq r(t_1, \dots, t_n)} \text{ (rule)}$$

$$\frac{\phi : s \geq t \quad \psi : t \geq u}{(\phi \cdot \psi) : s \geq u} \text{ (trans)}$$

2.8. Examples of proof terms

If $\varrho : f(x) \to x$, $\varrho(f(a))$ and $f(\varrho(a))$ both witness $f(f(a)) \geq f(a)$:

$$\dfrac{\dfrac{\dfrac{}{a : a \geq a}\,\text{(replacement)}}{f(a) : f(a) \geq f(a)}\,\text{(replacement)}}{\varrho(f(a)) : f(f(a)) \geq f(a)}\,\text{(rule)}$$

$$\dfrac{\dfrac{\dfrac{}{a : a \geq a}\,\text{(replacement)}}{\varrho(a) : f(a) \geq a}\,\text{(rule)}}{f(\varrho(a)) : f(f(a)) \geq f(a)}\,\text{(replacement)}$$

No syntactic accident!

2.9. Running example: proof term

TRS with named rules

$$
\begin{aligned}
\varrho &: \quad a \longrightarrow b \\
\vartheta &: \quad f(x, b) \longrightarrow g(x) \\
\varsigma &: \quad g(b) \longrightarrow c
\end{aligned}
$$

Proof terms for reductions

$$
\begin{aligned}
\mathcal{R} &= \underline{f(\varrho, a) \cdot f(b, \varrho)} \cdot \vartheta(b) \cdot \varsigma \\
&: \quad f(\tilde{a}, a) \longrightarrow_{\phi_1} f(b, \overline{a}) \longrightarrow_{\phi_2} \underline{f}(b, \underline{b}) \longrightarrow_{\phi_3} \underline{g}(\underline{b}) \longrightarrow_{\phi_4} c \\
\mathcal{S} &= f(a, \varrho) \cdot \vartheta(a) \cdot g(\varrho) \cdot \varsigma \\
&: \quad f(a, \overline{a}) \longrightarrow_{\psi_1} \underline{f}(a, \underline{b}) \longrightarrow_{\psi_2} g(\tilde{a}) \longrightarrow_{\psi_3} \underline{g}(\underline{b}) \longrightarrow_{\psi_4} c
\end{aligned}
$$

Home Page

Title Page

Contents

Page 15 of 56

Go Back

Full Screen

Close

Quit

2.10. Common restrictions on proof terms

TRS with rule $\varrho : f(x, y) \to x$, constants a, b, c, d

- Multi-step: no transitivity
  $\varrho(\varrho(a, b), c) : f(f(a, b), c) \mathrel{\circ\!\!\!\!\longrightarrow} a$
- Parallel step: no nested rule symbols
  $f(\varrho(a, b), \varrho(c, d)) : f(f(a, b), f(c, d)) \mathrel{+\!\!\!\!\!\!\longrightarrow} f(a, c)$
- Ordinary step: exactly one rule symbol
- Reduction: concatenation of ordinary steps
  $f(\varrho(a, b), c) \cdot \varrho(a, c)$
- Term: no rule symbols, no transitivity
  $f(a, c)$

3. Permutation equivalence

Solution 16 *Equivalence generated by permutations on proof terms.*

Ex 17 *(idea)* $[2, 1, 0]$ *and* $[0, 1, 2]$ *are equivalent:*

$$[2, \underline{1, 0}] \cong [\underline{2, 0}, 1] \cong [0, \underline{2, 1}] \cong [0, 1, 2]$$

by repeatedly permuting adjacent members

3.1. Permutation identities

$$
\begin{aligned}
1 \cdot \phi &\approx \phi \\
\phi \cdot 1 &\approx \phi \\
(\phi \cdot \psi) \cdot \chi &\approx \phi \cdot (\psi \cdot \chi) \\
f(\vec{\phi}) \cdot f(\vec{\psi}) &\approx f(\phi_1 \cdot \psi_1, \ldots, \phi_n \cdot \psi_n) \\
\varrho(\phi_1, \ldots, \phi_n) &\approx l(\phi_1, \ldots, \phi_n) \cdot \varrho(t_1, \ldots, t_n) \\
\varrho(\phi_1, \ldots, \phi_n) &\approx \varrho(s_1, \ldots, s_n) \cdot r(\phi_1, \ldots, \phi_n)
\end{aligned}
$$

– first three : reduction identities (monoid)

– first four : structural identities ($\equiv$)

– all : permutation identities ($\cong$)

Problem 18 *How to decide equivalence?*

3.2. Structural equivalence example

$$\varrho : \; a \;\longrightarrow\; b$$

3.2. Structural equivalence example

$$\varrho : \; a \;\longrightarrow\; b$$

$$\begin{aligned} g(\varrho, a) \cdot g(b, \varrho) \;&:\; g(a, a) \geq g(b, b) \\ g(a, \varrho) \cdot g(\varrho, b) \;&:\; g(a, a) \geq g(b, b) \end{aligned}$$

$$\begin{aligned} g(\varrho, a) \cdot g(b, \varrho) &\equiv g(\varrho \cdot b, a \cdot \varrho) \\ &\equiv g(\varrho, \varrho) \\ &\equiv g(a \cdot \varrho, \varrho \cdot b) \\ &\equiv g(a, \varrho) \cdot g(\varrho, b) \end{aligned}$$

3.3. Deciding structural equivalence

Solution 19 *Completed structural identities*

$$
\begin{aligned}
1 \cdot x &\Rightarrow x \\
x \cdot 1 &\Rightarrow x \\
(x \cdot y) \cdot z &\Rightarrow x \cdot (y \cdot z) \\
f(x_1, \ldots, x_n) \cdot f(y_1, \ldots, y_n) &\Rightarrow f(x_1 \cdot y_1, \ldots, x_n \cdot y_n) \\
f(\vec{x}) \cdot (f(y_1, \ldots, y_n) \cdot z) &\Rightarrow f(x_1 \cdot y_1, \ldots, x_n \cdot y_n) \cdot z
\end{aligned}
$$

$$
\begin{aligned}
\overline{g(\varrho, a) \cdot g(b, \varrho)} &\Rightarrow g(\overline{\varrho \cdot b}, \overline{a \cdot \varrho}) \\
&\Rightarrow g(\varrho, \varrho) \\
&\Leftarrow g(\underline{a \cdot \varrho}, \underline{\varrho \cdot b}) \\
&\Leftarrow \underline{g(a, \varrho) \cdot g(\varrho, b)}
\end{aligned}
$$

3.4. Running example: permutation equivalence

$$\begin{aligned}
\varrho &: \quad a \longrightarrow b \\
\vartheta &: \quad f(x,b) \longrightarrow g(x) \\
\varsigma &: \quad g(b) \longrightarrow c
\end{aligned}$$

$$\begin{aligned}
\mathcal{R} &= \underline{f(\varrho,a)\cdot f(b,\varrho)}\cdot\vartheta(b)\cdot\varsigma \\
&\cong \underline{f(\varrho,\varrho)}\cdot\vartheta(b)\cdot\varsigma \\
&\cong f(a,\varrho)\cdot\underline{f(\varrho,b)\cdot\vartheta(b)}\cdot\varsigma \\
&\cong f(a,\varrho)\cdot\underline{\vartheta(\varrho)}\cdot\varsigma \\
&\cong f(a,\varrho)\cdot\vartheta(a)\cdot g(\varrho)\cdot\varsigma \\
&= \mathcal{S}
\end{aligned}$$

4. Standardization equivalence

Solution 20 *Complete permutation identities*

Rewrite rule $\varrho : l \to r$, proof terms $x_i : s_i \geq t_i$

$$\begin{aligned} l(\vec{x}) \cdot \varrho(t_1, \ldots, t_n) &\Rightarrow_{\mathsf{I}} \varrho(x_1, \ldots, x_n) \\ \varrho(x_1, \ldots, x_n) &\Rightarrow_{\mathsf{O}} \varrho(s_1, \ldots, s_n) \cdot r(x_1, \ldots, x_n) \end{aligned}$$

apply modulo structural equivalence
normal forms are standard reductions

Thm 21 *Standardization is complete modulo structural equivalence*

Difficult, but term rewriting techniques (critical pairs) available because of proof terms

4.1. Running example: standardization equivalence

$$\mathcal{R} \quad : \quad f(\tilde{a}, a) \to f(b, \overline{a}) \to \underline{f}(b, \underline{b}) \to \underline{g}(\underline{b}) \to c$$

4.1. Running example: standardization equivalence

$$\mathcal{R} \ : \ f(\tilde{a}, a) \rightarrow f(b, \overline{a}) \rightarrow \underline{f}(b, \underline{b}) \rightarrow \underline{g}(\underline{b}) \rightarrow c$$

$$\equiv$$

4.1. Running example: standardization equivalence

$$
\begin{aligned}
\mathcal{R} \ &: \ f(\tilde{a}, a) \rightarrow f(b, \overline{a}) \rightarrow \underline{f}(b, \underline{b}) \rightarrow \underline{g}(\underline{b}) \rightarrow c \\
\equiv \ \mathcal{P} &: \ f(a, \overline{a}) \rightarrow f(\tilde{a}, b) \rightarrow \underline{f}(b, \underline{b}) \rightarrow \underline{g}(\underline{b}) \rightarrow c
\end{aligned}
$$

4.1. Running example: standardization equivalence

$$\begin{aligned}
\mathcal{R} \quad : \quad & f(\tilde{a}, a) \to f(b, \overline{a}) \to \underline{f}(b, \underline{b}) \to \underline{g}(\underline{b}) \to c \\
\equiv \ \mathcal{P} : \ & f(a, \overline{a}) \to f(\tilde{a}, b) \to \underline{f}(b, \underline{b}) \to \underline{g}(\underline{b}) \to c \\
\Rightarrow &
\end{aligned}$$

4.1. Running example: standardization equivalence

$$
\begin{aligned}
\mathcal{R} \quad &: \quad f(\tilde{a}, a) \to f(b, \overline{a}) \to \underline{f}(b, \underline{b}) \to \underline{g}(\underline{b}) \to c \\
\equiv \ \mathcal{P} &: \ f(a, \overline{a}) \to f(\tilde{a}, b) \to \underline{f}(b, \underline{b}) \to \underline{g}(\underline{b}) \to c \\
\Rightarrow \ \mathcal{S} &: \ f(a, \overline{a}) \to \underline{f}(a, \underline{b}) \to g(\tilde{a}) \to \underline{g}(\underline{b}) \to c
\end{aligned}
$$

$\mathcal{S}$ is standard

5. Labelling equivalence

Problem 22 *What is labelling?*

Exs 23 1. *Ornithology: ringing birds*

2. *λ-calculus: Church-typing for typable λ-terms*

3. *Term rewriting: semantic labelling of a TRS*

4. *Chemistry: label reaction by some (stable) isotope:*

$$CH_3COOH + C_2H_5OH \rightarrow C_2H_5COOCH_3 + H_2O$$

Label the second O as ^{18}O, giving $CH_3CO^{18}OH$:

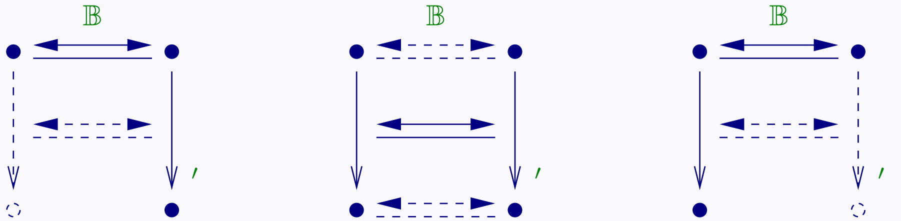
$$CH_3CO^{18}OH + C_2H_5OH \rightarrow C_2H_5COOCH_3 + H_2{}^{18}O$$

Solution 24 *Labelling: adjoining information in a behavior preserving way*

Home Page
Title Page
Contents
Page 24 of 56
Go Back
Full Screen
Close
Quit

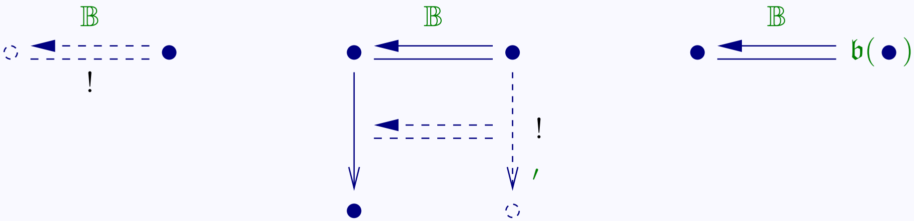
5.1. Bisimulation and labelling

Bisimulation



both objects and steps are related

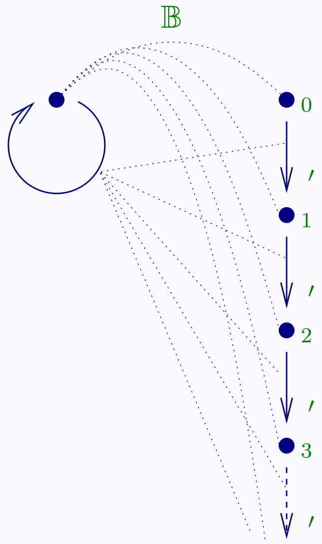
Labelling



labelling steps proceeds deterministically

5.2. Example: bisimulation/labelling

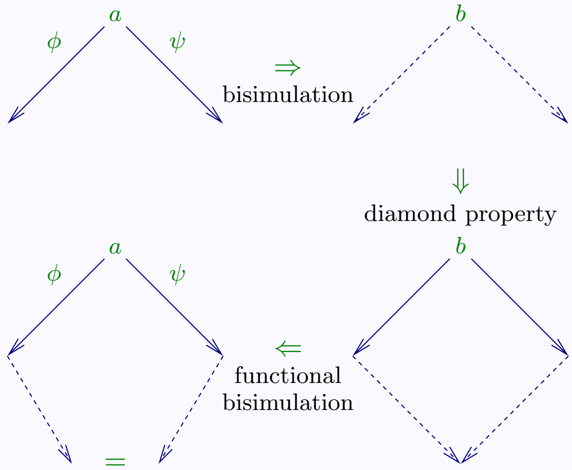
A bisimulation $\mathbb{B}$ between the black hole ARS $\circlearrowright$ and the infinite straight line ARS $-\!\infty\!\!\rightarrow$:



5.3. Transfer of properties along bisimulation/labelling

if $a \mathbb{B} b$ for some bisimulation $\mathbb{B}$, then

- a is normalizing (WN) iff b is normalizing,
- a is terminating (SN) iff b is terminating
- if $\mathbb{B}$ is a labelling, $\Diamond(b)$ implies $\Diamond(a)$



5.4. Term rewrite labelling

Term rewrite labelling $\mathfrak{B}$:

– Labelling $\mathbb{B}$ on its alphabets and rules

– initial labelling $\mathfrak{b}$ mapping terms to labelled terms

5.4. Term rewrite labelling

Term rewrite labelling $\mathfrak{B}$:

– Labelling $\mathbb{B}$ on its alphabets and rules

– initial labelling $\mathfrak{b}$ mapping terms to labelled terms

Stacks of natural numbers

rules push^n : $\top \to n(\top)$, pop^n : $n(\top) \to \top$

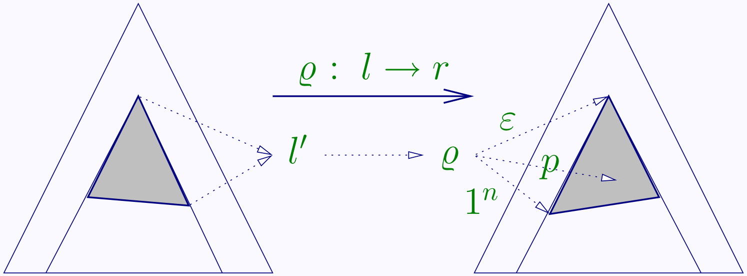
reduction $\top \to 5(\top) \to 5(1(\top)) \to 5(\top) \to 5(3(\top))$

label top of stack by height

rules push$_i^n$: $\top_i \to n(\top_{i+1})$, pop$_i^n$: $n(\top_{i+1}) \to \top_i$

reduction $\top_0 \to 5(\top_1) \to 5(1(\top_2)) \to 5(\top_1) \to 5(3(\top_2))$

5.5. Lévy labelling



record history of lhs in each symbol of rhs

rule: $\varrho : \; a \longrightarrow a$

reduction: $\mathcal{R} : \; a \longrightarrow a \longrightarrow a \longrightarrow a \longrightarrow \cdots$

Lévy labelled rules: $\varrho_{a^\varepsilon} : \; a^\varepsilon \longrightarrow \varrho^\varepsilon_{a^\varepsilon}$, $\varrho_{\varrho^\varepsilon_{a^\varepsilon}} : \; \varrho^\varepsilon_{a^\varepsilon} \longrightarrow \varrho^\varepsilon_{\varrho^\varepsilon_{a^\varepsilon}}$

Lévy labelled reduction: $a^\varepsilon \longrightarrow \varrho^\varepsilon_{a^\varepsilon} \longrightarrow \varrho^\varepsilon_{\varrho^\varepsilon_{a^\varepsilon}} \longrightarrow \varrho^\varepsilon_{\varrho^\varepsilon_{\varrho^\varepsilon_{a^\varepsilon}}} \longrightarrow \cdots$.

5.6. Labelling equivalence

Labelling equivalence: targets the same after any labelling

$$\begin{array}{lcl}\mathcal{R} & : & f(\tilde{a},a) \to f(b,\overline{a}) \to \underline{f}(b,\underline{b}) \to \underline{g}(\underline{b}) \to c \\ \mathcal{S} & : & f(a,\overline{a}) \to \underline{f}(a,\underline{b}) \to g(\tilde{a}) \to \underline{g}(\underline{b}) \to c\end{array}$$

Lévy labelling gives $\mathfrak{L}(\mathcal{R})$ and $\mathfrak{L}(\mathcal{S})$

$$f(\tilde{a},a) \to f(b^{\tilde{a}},a) \to f(b^{\tilde{a}},b^{a}) \to g^{f(x,b^{a})}(b^{\tilde{a}}) \to c^{g^{f(x,b^{a})}(b^{\tilde{a}})}$$

$$f(\tilde{a},a) \to f(\tilde{a},b^{a}) \to g^{f(x,b^{a})}(\tilde{a}) \to g^{f(x,b^{a})}(b^{\tilde{a}}) \to c^{g^{f(x,b^{a})}(b^{\tilde{a}})}$$

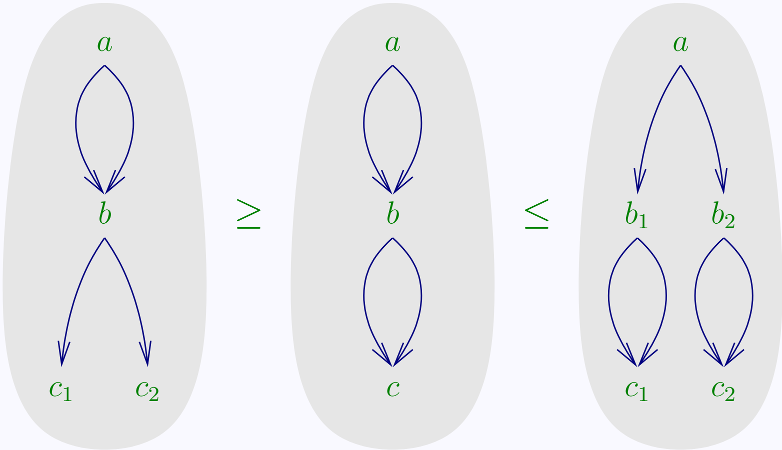
targets have same labelling, hence labelling equivalent

Note a not labelling equivalent to $a \to a$!

Home Page
Title Page
Contents
Page 30 of 56
Go Back
Full Screen
Close
Quit

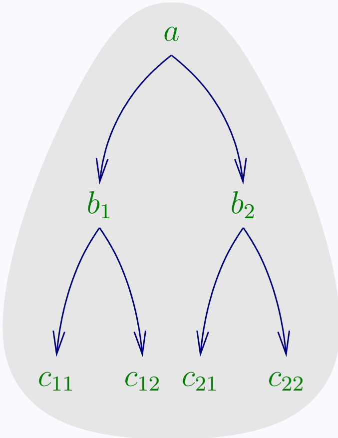
5.7. History order

History order $\mathfrak{A} \leq \mathfrak{B}$ if exists $\mathfrak{C}$, such that $\mathfrak{B}$ is the composition of $\mathfrak{A}$ and $\mathfrak{C}$



5.8. Maximality of Lévy labelling

Maximal with respect to history order.



Thm 25 *Lévy labelling is maximal among all term rewriting labellings*

Proof by tracing a connexion property

6. Projection equivalence

Problem 26 *What is projection?*

Solution 27 *Taking the residual*

$[2, 1, 0]$ and $[0, 1, 2]$ equivalent?

Home Page
Title Page
Contents
Page 32 of 56
Go Back
Full Screen
Close
Quit

6. Projection equivalence

Problem 26 *What is projection?*

Solution 27 *Taking the residual*

$[2, 1, 0]$ and $[0, 1, 2]$ equivalent?

$[0, 1, 2] \rightsquigarrow_2 [0, 1, \not{2}] \rightsquigarrow_1 [0, \not{1}, \not{2}] \rightsquigarrow_0 [\not{0}, \not{1}, \not{2}]$

$[2, 1, 0] \rightsquigarrow_0 [2, 1, \not{0}] \rightsquigarrow_1 [2, \not{1}, \not{0}] \rightsquigarrow_2 [\not{2}, \not{1}, \not{0}]$

6. Projection equivalence

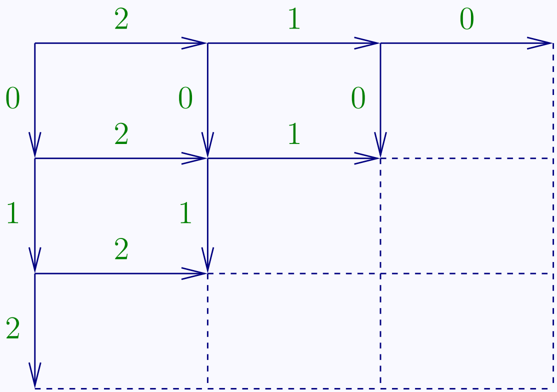
Problem 26 *What is projection?*

Solution 27 *Taking the residual*

$[2,1,0]$ and $[0,1,2]$ equivalent?

$[0,1,2] \rightsquigarrow_2 [0,1,\cancel{2}] \rightsquigarrow_1 [0,\cancel{1},\cancel{2}] \rightsquigarrow_0 [\cancel{0},\cancel{1},\cancel{2}]$

$[2,1,0] \rightsquigarrow_0 [2,1,\cancel{0}] \rightsquigarrow_1 [2,\cancel{1},\cancel{0}] \rightsquigarrow_2 [\cancel{2},\cancel{1},\cancel{0}]$



6.1. Examples of residual systems

Exs 28 *1. Natural numbers with cutoff-subtraction*

2. Sets with set-difference

3. Multiset with multiset difference

4. Braids

5. Stack numbers

6. Term rewriting: multi-step, parallel steps, proof terms (not ordinary steps, because of replication)

6.2. Residual Systems

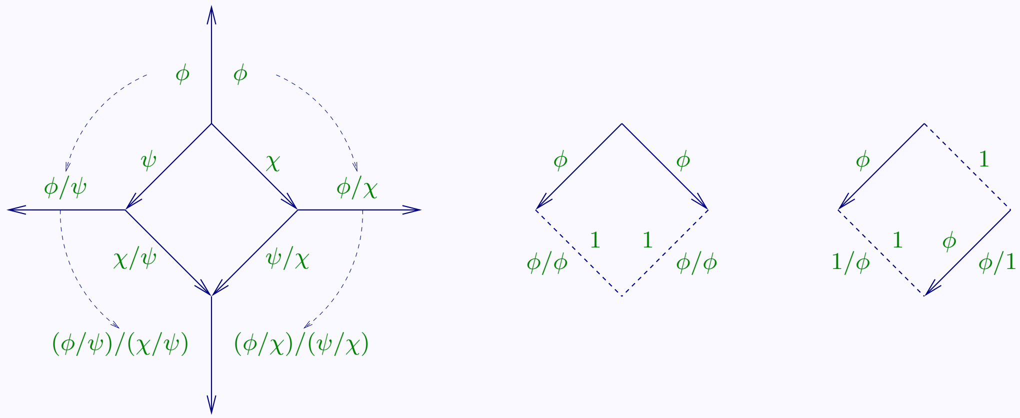
Def 29 *A residual system is a triple* $\langle \rightarrow, 1, / \rangle$*:*

- $\rightarrow$ *is an abstract rewriting system*
- 1 *(unit) is a function from objects to steps such that* $\mathsf{tgt}(1_a) = a = \mathsf{src}(1_a)$,
- $/$ *(residuation) is a function from pairs of co-initial steps to steps such that* $\mathsf{tgt}(\phi) = \mathsf{src}(\psi/\phi)$ *and* $\mathsf{tgt}(\phi/\psi) = \mathsf{tgt}(\psi/\phi)$

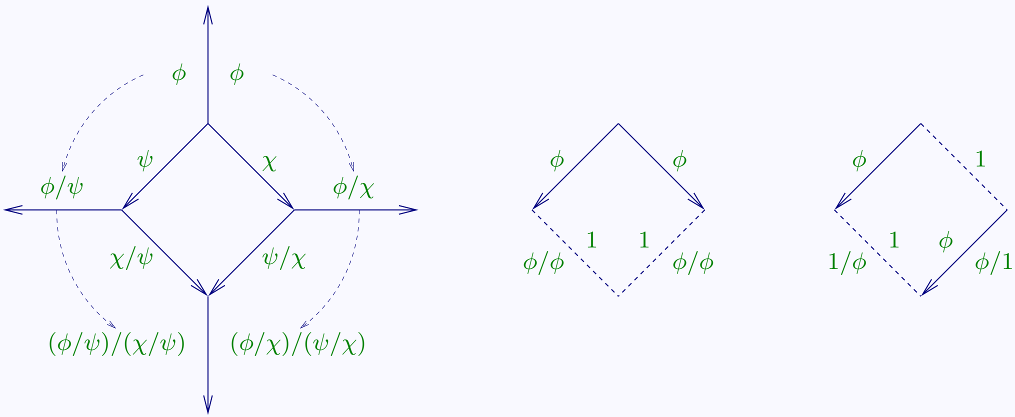
satisfying the residual identities:

$$
\begin{array}{rcl}
(\phi/\psi)/(\chi/\psi) & \approx & (\phi/\chi)/(\psi/\chi) \\
\phi/\phi & \approx & 1 \\
\phi/1 & \approx & \phi \\
1/\phi & \approx & 1
\end{array}
$$

6.3. Residual identities



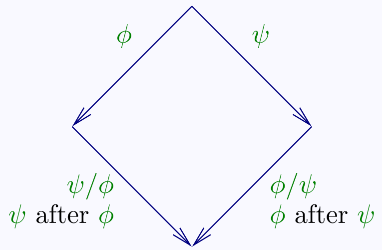
6.3. Residual identities

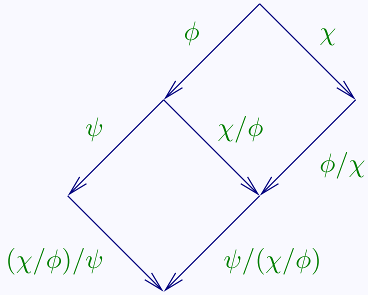


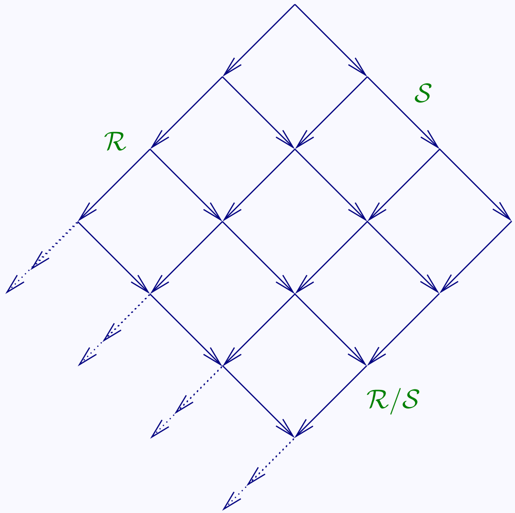
Ex 30 $1/\phi = (1/1)/(\phi/1) = (1/\phi)/(1/\phi) = 1$

Home Page
Title Page
Contents
Page 36 of 56
Go Back
Full Screen
Close
Quit

6.4. Residuals of steps, composites and reductions







6.5. Projection order

The projection order $\lesssim$ and the corresponding projection equivalence $\simeq$ are defined by

$$\phi \lesssim \psi \quad \text{if } \phi/\psi = 1$$
$$\phi \simeq \psi \quad \text{if } \phi \lesssim \psi \text{ and } \psi \lesssim \phi$$

for co-initial steps ϕ and ψ

projection order is quasi-order (need not be partial order!)

residuation monotonic in first (not necessarily antitonic in second)

projection equivalence congruence for the operations

Home Page

Title Page

Contents

Page 38 of 56

Go Back

Full Screen

Close

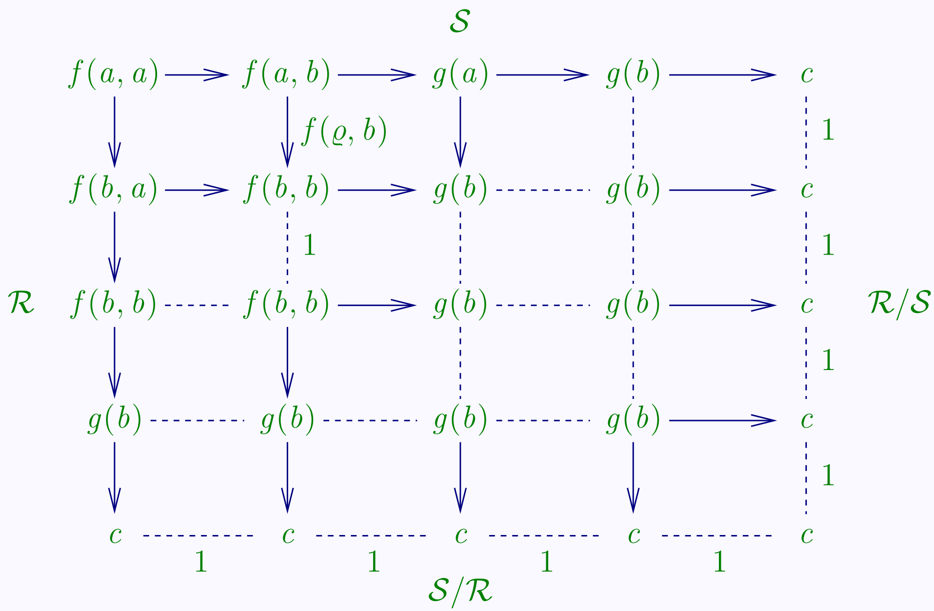
Quit

6.6. Term Residual Systems

Residual operation $/$:

$$
\begin{aligned}
f(\vec{\phi})/f(\vec{\psi}) &= f(\phi_1/\psi_1, \ldots, \phi_n/\psi_n) \\
\varrho(\vec{\phi})/l(\vec{\psi}) &= \varrho(\phi_1/\psi_1, \ldots, \phi_n/\psi_n) \\
l(\vec{\phi})/\varrho(\vec{\psi}) &= r(\phi_1/\psi_1, \ldots, \phi_n/\psi_n) \\
\varrho(\vec{\phi})/\varrho(\vec{\psi}) &= r(\phi_1/\psi_1, \ldots, \phi_n/\psi_n) \\
\chi/(\phi \cdot \psi) &= (\chi/\phi)/\psi \\
(\phi \cdot \psi)/\chi &= \phi/\chi \cdot \psi/(\chi/\phi) \\
\phi/\psi &= \mathsf{tgt}(\phi) \to_{\#} \# \quad \text{otherwise}
\end{aligned}
$$

6.7. Running example: projection equivalence



$$\mathcal{R} \; : \; f(\tilde{a},a) \to f(b,\overline{a}) \to \underline{f}(b,\underline{b}) \to \underline{g}(\underline{b}) \to c$$

$$\mathcal{S} \; : \; f(a,\overline{a}) \to \underline{f}(a,\underline{b}) \to g(\tilde{a}) \to \underline{g}(\underline{b}) \to c$$

Home Page

Title Page

Contents

Page 40 of 56

Go Back

Full Screen

Close

Quit

6.8. Running example: projection equivalence (ctd)

$$
\begin{aligned}
\mathcal{R}/\underline{\mathcal{S}} &= \underline{\mathcal{R}/(f(a,\varrho)\cdot\mathcal{S}_1)}\\
&= (\underline{\mathcal{R}}/f(a,\varrho))/\mathcal{S}_1\\
&= \underline{(f(\varrho,a)\cdot\mathcal{R}_1)/f(a,\varrho)}/\mathcal{S}_1\\
&= (\underline{f(\varrho,a)/f(a,\varrho)}\cdot(\mathcal{R}_1/\underline{f(a,\varrho)/f(\varrho,a)}))/\mathcal{S}_1\\
&= (f(\underline{\varrho/a},\underline{a/\varrho})\cdot(\mathcal{R}_1/f(\underline{a/\varrho},\underline{\varrho/a})))/\mathcal{S}_1\\
&= (f(\varrho,b)\cdot(\underline{\mathcal{R}_1}/f(b,\varrho)))/\mathcal{S}_1\\
&= (f(\varrho,b)\cdot\underline{(f(b,\varrho)\cdot\mathcal{R}_2)/f(b,\varrho)})/\mathcal{S}_1\\
&= (f(\varrho,b)\cdot\underline{f(b,\varrho)/f(b,\varrho)}\cdot(\mathcal{R}_2/\underline{f(b,\varrho)/f(b,\varrho)}))/\mathcal{S}_1\\
&= (f(\varrho,b)\cdot 1\cdot\underline{\mathcal{R}_2/1})/\mathcal{S}_1\\
&= (f(\varrho,b)\cdot 1\cdot\mathcal{R}_2)/\mathcal{S}_1
\end{aligned}
$$

Home Page

Title Page

Contents

Page 41 of 56

Go Back

Full Screen

Close

Quit

6.9. Well-definedness of projection

Termination non-obvious because calls itself, e.g.

$$(\phi \cdot \psi)/\chi = \phi/\chi \cdot \psi/(\chi/\phi)$$

Confluence non-obvious because of critical pairs, e.g.

$$((x \cdot y)/z)/w \Leftarrow (x \cdot y)/(z \cdot w) \Rightarrow (x/(z \cdot w)) \cdot (y/((z \cdot w)/x)$$

Solution: semantic labelling

6.10. Laws for term residual systems

$$
\begin{array}{rcl@{\qquad}rcl}
\phi/\phi & = & 1 & (\phi/\psi)/(\chi/\psi) & = & (\phi/\chi)/(\psi/\chi) \\
\phi/1 & = & \phi & \chi/(\phi\cdot\psi) & = & (\chi/\phi)/\psi \\
1/\phi & = & 1 & (\phi\cdot\psi)/\chi & = & (\phi/\chi)\cdot(\psi/(\chi/\phi)) \\
\phi\cdot 1 & \simeq & \phi & (\phi\cdot\psi)\cdot\chi & \simeq & \phi\cdot(\psi\cdot\chi) \\
1\cdot\phi & \simeq & \phi & \phi\sqcup\psi & = & \phi\cdot(\psi/\phi) \\
1\sqcup\phi & \simeq & \phi & \chi/(\phi\sqcup\psi) & = & (\chi/\phi)/(\psi/\phi) \\
\phi\sqcup\phi & \simeq & \phi & (\phi\sqcup\psi)/\chi & = & (\phi/\chi)\sqcup(\psi/\chi) \\
\phi\sqcup\psi & \simeq & \psi\sqcup\phi & \phi\cdot(\psi\sqcup\chi) & \simeq & (\phi\cdot\psi)\sqcup(\phi\cdot\chi) \\
(\phi\sqcup\psi)\sqcup\chi & \simeq & \phi\sqcup(\psi\sqcup\chi) & \phi\cdot\psi\simeq\phi\cdot\chi & \Rightarrow & \psi\simeq\chi
\end{array}
$$

Join $\phi \sqcup \psi$ is defined as $\phi \cdot (\psi/\phi)$

7. Equivalences are the same

Problem 31 *How to show correspondence between permutation and labelling?*

Solution 32
- *labelling preserves permutation equivalence*
- *Lévy labelling allows for reconstruction up to permutation equivalence*

7.1. Reconstruction for ARSs

Unwinding of an ARS: record the complete history in the objects

Thm 33 (Reconstruction Theorem) *There exists a reconstruction map* $\mathfrak{R}$ *from unwound objects to ordinary reductions such that for any reduction* $\mathcal{R}$ *and any object* a' *in the unwinding*

$$\mathfrak{R}(a') = \mathcal{R} \textit{ iff } a' = \mathsf{tgt} \circ \mathfrak{U}(\mathcal{R})$$

Unwinding : ARS as Lévy labelling : TRS

7.2. Reconstruction for TRSs

Problem 34 *For term rewriting systems:*

- *order of steps is not recorded*
- *erased parts cannot be reconstructed*

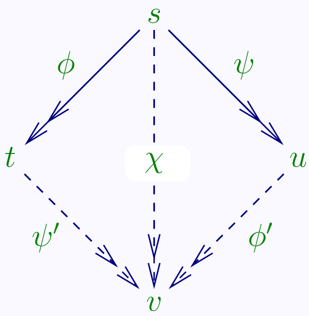
Solution 35 *Reconstruct needed prefix up to permutation*

Thm 36 *There exists a reconstruction function* $\mathfrak{R}$ *which has the property that for any extracted reduction* $\mathcal{R} \triangleright Q$ *and any Lévy labelling* Q' *of its target*

$$\mathfrak{R}(Q') = \mathcal{R} \triangleright Q \quad \textit{iff} \quad Q' = \mathrm{tgt} \circ \mathfrak{L}(\mathcal{R} \triangleright Q) \qquad (1)$$

Home Page
Title Page
Contents
Page 47 of 56
Go Back
Full Screen
Close
Quit

8. Parallel Proofs Lemma for orhogonal TRSs



for given coinitial ϕ, ψ there exist proof terms $\psi' : t \geq v$, $\chi : s \geq v$ and $\phi' : u \geq v$ without error symbols, such that

1. The common reduct v is found via permutation equivalent ways, $\phi \cdot \psi' \cong \chi \cong \psi \cdot \phi'$, and χ is a minimal proof term in the permutation order having this property.

2. The targets of the Lévy labellings (formally: liftings) of the ways are identical, $\mathsf{tgt}(\mathfrak{L}(\phi \cdot \psi')) = \mathsf{tgt}(\mathfrak{L}(\chi)) = \mathsf{tgt}(\mathfrak{L}(\psi \cdot \phi'))$, and $\mathsf{tgt}(\mathfrak{L}(\chi))$ rewrites to any other Lévy labelled term having this property.

3. The ways $\phi \cdot \psi'$, χ and $\psi \cdot \phi'$ yield identical (parallel) standard reductions.

4. The three ways can be computed using the residual and join operations of the associated residual system: $\psi' \simeq \psi / \phi$, $\chi \simeq \phi \sqcup \psi$ and $\phi' \simeq \phi / \psi$.

9. λ-calculus with end-of-scope

Observations

- λx is like opening an 'x-bracket'
- no corresponding closing bracket!

Proposal

- Adjoin closing 'x-brackets' $\reflectbox{$\lambda$}x$
  (adbmal, unbind, end-of-scope)
- $\reflectbox{$\lambda$}x.M$ closes matching λx: x is 'free' in $\lambda x.\reflectbox{$\lambda$}x.x$
- Can be nested $\underbrace{\lambda x.\overbrace{\lambda x.\reflectbox{λ}x}.\reflectbox{λ}x}.x$ (x free again)
- Proper nesting: $\lambda x.\lambda y.\reflectbox{λ}x.\reflectbox{λ}y.M$ not allowed
  (better: $\reflectbox{$\lambda$}x$ implictly closes λy)

10. β-reduction

Observation: $\not\lambda$'s in between $@$ and λ

- $(\lambda x.\not\lambda x.M)N$ should reduce to M
- $(\not\lambda y.\lambda x.x)y$ should not reduce to $\not\lambda y.y$ (but y)
- $(\not\lambda y.\lambda x.z)y$ should not reduce to z (but $\not\lambda y.z$)

Where should end-of-scopes go?

- search for matching x
- in case of x: remove end-of-scopes, put argument
- in case of $\not\lambda x$: put end-of-scopes, remove argument

$(\not\lambda X.\lambda x.M)N \to M[X, x{:=}N, \Box]$:

- X remembers the end-of-scopes
- third argument: stack used for matching

11. Formalization in Coq

Axiom 1 *Assume a parameter set* $\mathcal{V}$ *of infinitely many variable names for which equality is decidable.*

– $x = y \vee x \neq y$ *for all* $x, y : \mathcal{V}$

– $\exists x : \mathcal{V}.\, x \notin X$ *for all* $X : \mathit{list}(\mathcal{V})$

```
Parameter name : Set.
Axiom eq_dec : (x,y:name){x=y}+{~x=y}.
Axiom inf_many_names :
  (l:(list name)){a:name|~(In a l)}.
```

Def 37 *Substitution* $M[X, x{:=}N, Y]$ *is defined by:*

$$
\begin{aligned}
y[X, x{:=}N, Y] &= y, \text{ if } y \in Y \\
y[X, x{:=}N, Y] &= \reflectbox{λ} Y.N, \text{ if } y \notin Y, x = y \\
y[X, x{:=}N, Y] &= \reflectbox{λ} Y.\reflectbox{λ} X.y, \text{ if } y \notin Y, x \neq y \\
(\lambda y.M)[X, x{:=}N, Y] &= \lambda y.M[X, x{:=}N, yY] \\
(\reflectbox{λ} y.M)[X, x{:=}N, \Box] &= \reflectbox{λ} X.M, \text{ if } x = y \\
(\reflectbox{λ} y.M)[X, x{:=}N, \Box] &= \reflectbox{λ} X.\reflectbox{λ} y.M, \text{ if } x \neq y \\
(\reflectbox{λ} y.M)[X, x{:=}N, zY] &= \reflectbox{λ} y.M[X, x{:=}N, Y], \text{ if } y = z \\
(\reflectbox{λ} y.M)[X, x{:=}N, zY] &= (\reflectbox{λ} y.M)[X, x{:=}N, Y], \text{ if } y \neq z \\
(M_1 M_2)[X, x{:=}N, Y] &= M_1[X, x{:=}N, Y] M_2[X, x{:=}N, Y]
\end{aligned}
$$

Lem 1 *Closed substitution lemma: if* $\langle X'xZ\rangle s$, $\langle Y'yZ'Z\rangle t$ *and* $\langle YZ'Z\rangle u$, *then:*

$$s[Y'yZ', x{:=}t, X'][Y, y{:=}u, X'Y'] \\ = s[Y'YZ', x{:=}t[Y, y{:=}u, Y'], X']$$

```
Lemma closed_subst_bal :
 (s,t,u:term;X',Y,Y',Z,Z':(list name);x,y:name)
 (bal (conc X' (cons x Z)) s)
  -> (bal (conc Y' (conc (cons y Z') Z)) t)
  -> (bal (conc Y (conc Z' Z)) u)
   -> (subst Y (conc X' Y') (subst (conc Y'
            (cons y Z')) X' s x t) y u)
     = (subst (conc Y' (conc Y Z')) X' s x
             (subst Y Y' t y u)).
```

Lem 2 *Open substitution lemma: if* $\langle X'xY'yZ\rangle s$, $\langle XY'yZ\rangle t$ *and* $\langle YZ\rangle u$, *then:*

$$s[X, x{:=}t, X'][Y, y{:=}u, X'XY'] = s[Y, y{:=}u, X'xY'][X, x{:=}t[Y, y{:=}u, XY'], X']$$

```
Lemma open_subst_bal :
 (s,t,u:term;X,X',Y,Y',Z:(list name);x,y:name)
  (bal (conc X' (conc (cons x Y')(cons y Z))) s)
   -> (bal (conc X (conc Y' (cons y Z))) t)
    -> (bal (conc Y Z) u)
     -> (subst Y (conc X' (conc X Y'))
         (subst X X' s x t) y u)
         = (subst X X' (subst Y
         (conc X' (cons x Y')) s y u) x
         (subst Y (conc X Y') t y u)).
```

Home Page

Title Page

Contents

Page 55 of 56

Go Back

Full Screen

Close

Quit

12. Results

- Confluence of β without α!
- α needed to remove $\not\lambda$'s:
  $\lambda x.\not\lambda x.x \rightarrow_{\alpha} \lambda y.\not\lambda y.x \rightarrow_{\mathsf{forget}} \lambda y.x$
- Confluence of ordinary β modulo α
- Proofs in Coq (also for de Bruijn indices)

Home Page

Title Page

Contents

◀◀ ▶▶

◀ ▶

Go Back

Full Screen

Close

Quit

13. Related/current work

Related work

– refines Guillaume, David, Bird, Patterson: de Bruijn ($\lambdabar$ = Successor, variable = zero)

– Kesner, Di Cosmo, ...: with labels (commutativity, α)

– Explicit weakening

Current work

– Push $\lambdabar$'s locally : $\lambdabar x.\lambda x.M \to \lambda_1 x.\lambdabar x.M$

– inverses (reopen scope...)

– optimal implementation (brackets, no scopes)

– explicit substitution (lemmas as rules) (CR, PSN)