Rewriting, Explicit Substitutions and Normalisation

XXXVI Escola de Verão do MAT

Universidade de Brasilia

Part 2/3

Eduardo Bonelli

LIFIA (Fac. de Informática, UNLP, Arg.) and CONICET

eduardo@lifia.info.unlp.edu.ar

February, 2006

Structure of Today's Talk

1. Residuals
2. Standardization
3. Needed Strategies

From hereon we work with left-linear TRS

1. Residuals
   - Examples and Definition
   - Equivalence of Derivations
2. Standardization
3. Needed Strategies

Example

Consider the TRS

$$\begin{array}{rlcl} \rho: & a & \rightarrow & b \\ \vartheta: & f(x,a) & \rightarrow & g(x,x) \end{array}$$

and the term

$$f(a,a)$$

It has three redexes, r, s and t:

f(a,a)   f(a,a)   f(a,a)

Example

$$\begin{array}{rlcl} \rho: & a & \to & b \\ \vartheta: & f(x,a) & \to & g(x,x) \end{array}$$

Consider the redexes r and s:

$$\mathrm{f(a,a)} \quad \mathrm{f(a,a)}$$

Reducing s leaves a "leftover" or *residual* of r

$$f(a,a) \to_{\rho} f(b,a)$$

Likewise reducing r leaves a "leftover" or *residual* (two actually) of s

$$f(a,a) \to_{\vartheta} g(a,a)$$

Note: r and s do *not* overlap

Example

$$\begin{array}{llll} \rho: & a & \longrightarrow & b \\ \vartheta: & f(x,a) & \longrightarrow & g(x,x) \end{array}$$

Consider the redexes r and t:

$$\mathrm{f(a,a)} \quad \mathrm{f(a,a)}$$

Reducing r leaves no residual of t; Reducing t leaves no residual of r

- Note: r and t overlap
- Only other case where a redex leaves no residual: when it is erased. Eg. replacing ϑ by $f(x,a) \to b$, note a is erased below

$$f(a,a) \to_{\vartheta} b$$

Definition of Residuals

Assume ρ-redex r and ϑ-redex s in M and $M \to_r N$

What happens with s after the r-step?

Consider all cases:

1. They are disjoint: s appears in N
2. They are equal: s is erased in N
3. s is in an argument of r: s appears $n \geq 0$ times in N
4. r is in an argument of s: s appears in N with a different argument
5. r and s overlap: s is erased

In general, there is no sense in defining the residual of a redex after an overlapping reduction step (case 5)

Residual relation

Let $r : M \to N$. The residual relation for r

$$\_/_{r}$$

is defined as above: it maps nonoverlapping redexes in M to the set of their residuals

Basic properties

1. $r/_{r} = \emptyset$
2. $s/_{r}$ is a finite set of redexes

Redex Creation

Let $r : M \to N$. Redexes in N that are not residuals of those in M are called created

$$\begin{array}{rlcl} \rho : & a & \to & b \\ \vartheta : & f(x,b) & \to & g(x,x) \end{array}$$

The redex $f(a,b)$ is created in

$$f(a,a) \to_\rho f(a,b)$$

Residual Relation on Derivations

The residual relation extends to derivations

$$u \in s/_{(r;d)} \text{ iff } \exists v \text{ s.t. } v \in s/_r \text{ and } u \in v/_d$$

Informally,

$$\begin{tikzcd}
M_1 \arrow[r, "r"'] \arrow[d, "s"'] & M_2 \arrow[r, two heads, "d"'] \arrow[d, "v"'] & M_n \arrow[d, "u"'] \\
{} & {} & {}
\end{tikzcd}$$

Multi-redex

Multi-redex is a pair $\langle M, U\rangle$ where U is a finite set of nonoverlapping redexes in M

Residual relation $/_d$ extends to multi-redexes:

$$\langle M, U\rangle /_d \langle N, V\rangle$$

when

1. $d : M \twoheadrightarrow N$
2. V is the set of residuals of elements of U after d
$$V = \{v \mid \exists u \in U,\ u /_d v\}$$

Development

A derivation

$$d : M = M_1 \to_{r_1} M_2 \to_{r_2} M_3 \ldots \to_{r_{n-1}} M_n \to_{r_n} M_{n+1}$$

develops a multi-redex $\langle M, U\rangle$ partially when every redex r_i is an element of the multi-redex

$$\langle M, U\rangle /_{r_1;\ldots;r_{i-1}}$$

We say $d : M \twoheadrightarrow M_{n+1}$ develops the multi-redex $\langle M, U\rangle$ when d develops $\langle M, U\rangle$ partially and

$$\langle M, U\rangle /_d = \langle M_{n+1}, \emptyset\rangle$$

Example

Consider the TRS

$$\begin{array}{lrcl} \rho: & a & \to & b \\ \vartheta: & f(x,b) & \to & g(x,x) \end{array}$$

and the multi-redex

$$\langle f(a,b), \{r,s\}\rangle$$

1. The derivation $f(a,b) \to f(b,b)$ partially develops this multi-redex
2. Both derivations below develop this multi-redex
$$f(a,b) \to f(b,b) \to g(b,b) \text{ and}$$
$$f(a,b) \to g(a,a) \to g(b,a) \to g(b,b)$$
3. For the multi-redex $\langle f(a,b), \{s\}\rangle$ the derivation
$$f(a,b) \to g(a,a)$$
is not a partial development

Finite Developments

Lemma

For every multi-redex $\langle M, U \rangle$, there does not exist any infinite derivation

$$M_1 \to_{r_1} M_2 \to_{r_2} M_3 \to_{r_3} \dots$$

s.t. for each i the derivation

$$M_1 \to_{r_1} M_2 \to_{r_2} M_3 \to_{r_3} \dots M_{i-1} \to_{r_{i-1}} M_i$$

develops $\langle M, U \rangle$ partially

Informally,

Contraction (only) of residuals of a fixed set U of redexes in M eventually terminates

Basic Tile

Lemma (Parallel Moves)

For every two coinitial, non-overlapping redexes $r : M \longrightarrow P$ and $s : M \longrightarrow Q$ there exists two derivations d_r and d_s s.t.

1. d_r develops $r/_s$ and d_s develops $s/_r$
2. d_r and d_s are cofinal and induce the same residual relation

$$
\begin{tikzcd}
M \arrow[r, "r"'] \arrow[d, "s"'] & P \arrow[d, two heads, "d_s"'] \\
Q \arrow[r, two heads, "d_r"'] & N
\end{tikzcd}
$$

Basic Tile and Equivalence of Derivations

- Basic tile provides a convenient mechanism for defining a notion of equivalence of derivations
- Intuition: $d : M \twoheadrightarrow N$ and $e : M \twoheadrightarrow N$ are "equivalent" if they do the same "work" but in different "order"

$$\begin{tikzcd}
f(a,a) \arrow[rr, "a"'] \arrow[d, "a"'] & & f(b,a) \arrow[d, "a"'] \\
f(a,b) \arrow[rr, "a"'] \arrow[d, "f(\Box{,}b)"'] & & f(b,b) \arrow[d, "f(\Box{,}b)"'] \\
g(a,a) \arrow[r, "a"'] & g(a,b) \arrow[r, "a"'] & g(b,b)
\end{tikzcd}$$

Lévy Permutation Equivalence

Write $f \equiv^1 g$ if $f = f_1; r; d_s; f_2$ and $g = f_1; s; d_r; f_2$ and the diagram below is a basic tile

$$
\begin{tikzcd}
R \arrow[r, twoheadrightarrow, "f_1"'] & M \arrow[r, "r"'] \arrow[d, "s"'] & P \arrow[d, twoheadrightarrow, "d_s"'] & \\
& Q \arrow[r, twoheadrightarrow, "d_r"'] & N \arrow[r, twoheadrightarrow, "f_2"'] & S
\end{tikzcd}
$$

Lévy permutation equivalence is the least equivalence relation on derivations containing $\equiv^1$

Informally,

- $f \equiv g$ if there is a finite sequence of basic tilings connectinf f and g

Lévy Permutation Equivalence - Example Revisited

$$
\begin{tikzcd}
f(a,a) \arrow[rr, "a"'] \arrow[d, "a"'] \arrow[drr, phantom, "\equiv"] & & f(b,a) \arrow[d, "a"'] \\
f(a,b) \arrow[rr, "a"'] \arrow[d, "f(\Box,b)"'] \arrow[drr, phantom, "\equiv"] & & f(b,b) \arrow[d, "f(\Box,b)"'] \\
g(a,a) \arrow[r, "a"'] & g(a,b) \arrow[r, "a"'] & g(b,b)
\end{tikzcd}
$$

Epimorphism

Lemma ([Berry] for λ)

$$d; e \equiv d; f \text{ implies } e \equiv f$$

All arrows are epi in the category generated by the reduction graph of an OTRS with $\equiv$ as identity on arrows

Algebraic Confluence

Thm ([Lévy1978] for λ)

Let $d : M \twoheadrightarrow P$ and $e : M \twoheadrightarrow Q$ be coinitial in an OTRS. Then:

$$
\begin{tikzcd}
M \arrow[r, two heads, "d"'] \arrow[d, two heads, "e"'] \arrow[dr, phantom, "\equiv"] & P \arrow[d, two heads, "e'"'] \arrow[ddr, bend left, two heads, "e''"] & \\
Q \arrow[r, two heads, "d'"'] \arrow[drr, bend right, two heads, "d''"'] & N \arrow[dr, dotted, two heads, "f"'] \arrow[r, phantom, "\equiv"] \arrow[d, phantom, "\equiv"] & \\
 & & O
\end{tikzcd}
$$

The category generated by the reduction graph of an OTRS with $\equiv$ as identity on arrows enjoys pushouts

Standardization

The idea: for any derivation from M to N, there is a canonical derivation from M to N that computes redexes "outside-in"

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

Not standard

$$f(a,b) \to f(b,b) \to g(b,b)$$

Standard

$$f(a,b) \to g(a,a) \to g(b,a) \to g(b,b)$$

Examples - Uniqueness

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

Both these derivations are standard

$$f(a,a) \to f(b,a) \to f(b,b)$$
$$f(a,a) \to f(a,b) \to f(b,b)$$

- They are essentially the same (compute disjoint redexes in different order)!
- We thus identify derivations differing in this inessential way

Reversible Permutation Equivalence

Write $f \simeq^1 g$ if $f = f_1; r; d_s; f_2$ and $g = f_1; s; d_r; f_2$ and r, s are disjoint and the diagram below is a basic tile

$$\begin{tikzcd}
R \arrow[r, two heads, "f_1"'] & M \arrow[r, "r"'] \arrow[d, "s"'] & P \arrow[d, "s'"'] & \\
 & Q \arrow[r, "r'"'] & N \arrow[r, two heads, "f_2"'] & S
\end{tikzcd}$$

Reversible permutation equivalence is the least equivalence relation on derivations containing $\simeq^1$

Informally,

- $f \simeq g$ if there is a finite sequence of swappings of disjoint redexes from f to g

Defining Standard Derivations Through Permutation

Example 1

$$
\begin{tikzcd}
f(a,b) \arrow[r, "a"'] & f(b,b) \arrow[d, "f(\Box,b)"'] \\
& g(b,b)
\end{tikzcd}
$$

Defining Standard Derivations Through Permutation

Example 1

$$\begin{tikzcd}
f(a,b) \arrow[r, "a"'] \arrow[d, "f(\square{,}b)"'] & f(b,b) \arrow[d, "f(\square{,}b)"'] \\
g(a,a) \arrow[r, twoheadrightarrow, "a"'] & g(b,b)
\end{tikzcd}$$

Defining Standard Derivations Through Permutation

Example 1

$$
\begin{tikzcd}
f(a,b) \arrow[r, "a"'] \arrow[d, "f(\Box{,}b)"'] & f(b,b) \arrow[d, "f(\Box{,}b)"'] \\
g(a,a) \arrow[r, twoheadrightarrow, "a"'] & g(b,b)
\end{tikzcd}
$$

Example 2

$$
\begin{tikzcd}
f(a,a) \arrow[r, "a"'] & f(b,a) \arrow[d, "a"'] \\
& f(b,b) \arrow[d, "f(\Box{,}b)"'] \\
& g(b,b)
\end{tikzcd}
$$

Defining Standard Derivations Through Permutation

Example 1

$$
\begin{tikzcd}
f(a,b) \arrow[r, "a"'] \arrow[d, "f(\Box,b)"'] & f(b,b) \arrow[d, "f(\Box,b)"'] \\
g(a,a) \arrow[r, two heads, "a"'] & g(b,b)
\end{tikzcd}
$$

Example 2

$$
\begin{tikzcd}
f(a,a) \arrow[r, "a"'] \arrow[d, "a"'] \arrow[dr, phantom, "\simeq"] & f(b,a) \arrow[d, "a"'] \\
f(a,b) \arrow[r, "a"'] & f(b,b) \arrow[d, "f(\Box,b)"'] \\
 & g(b,b)
\end{tikzcd}
$$

Defining Standard Derivations Through Permutation

Example 1

$$\begin{array}{ccc} f(a,b) & \xrightarrow[a]{} & f(b,b) \\ {\scriptstyle f(\Box,b)}\downarrow & & \downarrow{\scriptstyle f(\Box,b)} \\ g(a,a) & \xrightarrow[a]{} & g(b,b) \end{array}$$

Example 2

$$\begin{array}{ccc} f(a,a) & \xrightarrow[a]{} & f(b,a) \\ {\scriptstyle a}\downarrow & \simeq & \downarrow{\scriptstyle a} \\ f(a,b) & \xrightarrow[a]{} & f(b,b) \\ {\scriptstyle f(\Box,b)}\downarrow & & \downarrow{\scriptstyle f(\Box,b)} \\ g(a,a) & \xrightarrow[a]{} & g(b,b) \end{array}$$

Standardizing Permutation

We need one more ingredient for defining standard derivations

There is a standardizing permutation from $f : M \twoheadrightarrow N$ to $g : M \twoheadrightarrow N$ (written $f \Rightarrow g$) iff

1. $f = f_1; r; d_s; f_2$ and $g = f_1; s; d_r; f_2$
2. s nests r and
3. the diagram below is a basic tile

$$\begin{tikzcd} R \arrow[r, two heads, "f_1"'] & M \arrow[r, "r"'] \arrow[d, "s"'] & P \arrow[d, two heads, "d_s"'] & \\ & Q \arrow[r, two heads, "d_r"'] & N \arrow[r, two heads, "f_2"'] & S \end{tikzcd}$$

In fact, since s nests r, d_s will consist of just one redex
So we can write this definition more accurately as follows

Standardizing Permutation

We need one more ingredient for defining standard derivations

There is a standardizing permutation from $f : M \twoheadrightarrow N$ to $g : M \twoheadrightarrow N$ (written $f \Rightarrow g$) iff

1. $f = f_1; r; d_s; f_2$ and $g = f_1; s; d_r; f_2$
2. s nests r and
3. the diagram below is a basic tile

$$\begin{tikzcd}
R \arrow[r, two heads, "f_1"'] & M \arrow[r, "r"'] \arrow[d, "s"'] & P \arrow[d, two heads, "d_s"'] & \\
& Q \arrow[r, two heads, "d_r"'] & N \arrow[r, two heads, "f_2"'] & S
\end{tikzcd}$$

In fact, since s nests r, d_s will consist of just one redex
So we can write this definition more accurately as follows

Standardizing Permutation

There is a standardizing permutation from $f : M \twoheadrightarrow N$ to $g : M \twoheadrightarrow N$ (written $f \Rightarrow g$) iff

1. $f = f_1; r; s'; f_2$ and $g = f_1; s; d_r; f_2$
2. s nests r and
3. the diagram below is a basic tile

$$\begin{tikzcd}
R \arrow[r, twoheadrightarrow, "f_1"'] & M \arrow[r, "r"'] \arrow[d, "s"'] & P \arrow[d, "s'"'] & \\
& Q \arrow[r, twoheadrightarrow, "d_r"'] & N \arrow[r, twoheadrightarrow, "f_2"'] & S
\end{tikzcd}$$

Standard Derivation

A derivation f is standard if there is no derivation g s.t.

$$[f]_{\simeq} \Rightarrow [g]_{\simeq}$$

Note: $[f]_{\simeq}$ is the reversible permutation equivalence class of f

Informally,

f is standard if no disjoint permutation of redexes of f gives rise to a standardizing permutation.

Example 1 - Revisited

$$\begin{array}{lcl} a & \rightarrow & b \\ f(x,b) & \rightarrow & g(x,x) \end{array}$$

The derivation $d : f(a,b) \rightarrow f(b,b) \rightarrow g(b,b)$ is not standard

$$\begin{tikzcd}
f(a,b) \arrow[r, "a"'] \arrow[d, "f(\Box{,}b)"'] & f(b,b) \arrow[d, "f(\Box{,}b)"'] \\
g(a,a) \arrow[r, twoheadrightarrow, "a"'] & g(b,b)
\end{tikzcd}$$

Indeed,

$$d \Rightarrow f(a,b) \rightarrow g(a,a) \rightarrow g(b,a) \rightarrow g(b,b)$$

Example 2 - Revisited

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

The derivation $d : f(a,a) \to f(b,a) \to f(b,b) \to g(b,b)$ is not standard

$$\begin{tikzcd}
f(a,a) \arrow[r, "a"'] \arrow[d, "a"'] \arrow[dr, phantom, "\simeq"] & f(b,a) \arrow[d, "a"] \\
f(a,b) \arrow[r, "a"'] \arrow[d, "f(\Box,b)"'] \arrow[dr, phantom, "\Leftarrow"] & f(b,b) \arrow[d, "f(\Box,b)"] \\
g(a,a) \arrow[r, twoheadrightarrow, "a"'] & g(b,b)
\end{tikzcd}$$

Indeed, $$\begin{array}{rcl} d & \simeq & f(a,a) \to f(a,b) \to f(b,b) \to g(b,b) \\ & \Rightarrow & f(a,a) \to f(a,b) \to g(a,a) \to g(b,a) \to g(b,b) \end{array}$$

Standardization Theorem

Thm

1. Existence
   For every $d : M \twoheadrightarrow N$, there exists a standard derivation $\mathsf{std}(d) : M \twoheadrightarrow N$ and $d_1, \ldots, d_n$ s.t.
   $$[d]_\simeq \Rightarrow [d_1]_\simeq \Rightarrow \ldots \Rightarrow [d_n]_\simeq \Rightarrow [\mathsf{std}(d)]_\simeq$$
2. Uniqueness
   Let $d, e : M \twoheadrightarrow N$ s.t. $d \equiv e$ (i.e. d and e are Lévy permutation equivalent). Then
   $$[\mathsf{std}(d)]_\simeq = [\mathsf{std}(e)]_\simeq$$

Computing Standard Derivations

1. Repeatedly apply standardizing permutations on $\simeq$-equivalence classes
   - See [Terese, Sec.8.5.3 ("Inversion Parallel Standardization")] where this process is shown to be CR and SN
2. Alternative standardization procedure
   - Given $d : M \twoheadrightarrow N$ compute $\mathrm{std}(d)$ by repeatedly extracting outermost redexes contracted in d
   - This yields a standard derivation $\equiv$-equivalent to d
   - See [Terese, Sec.8.5.2. ("Selection Parallel Standardization")]

Reduction Strategy

A (one-step or many step) reduction strategy for a TRS $\mathcal{R}$ is a function $\mathbb{F} : \mathcal{T}(\Sigma) \to \mathcal{T}(\Sigma)$ s.t.

1. $\mathbb{F}(M) = M$, if M is in $\mathcal{R}$-normal form
2. $M \to^+ \mathbb{F}(M)$, otherwise

$\mathbb{F}$ is normalizing iff for every WN term M there is no infinite reduction sequence

$$M \to^+ \mathbb{F}(M) \to^+ \mathbb{F}(\mathbb{F}(M)) \to^+ \mathbb{F}(\mathbb{F}(\mathbb{F}(M))) \to^+ \ldots$$

Example

Consider the TRS

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

$f(f(a, f(a, b)), g(f(a, b), g(b, a)))$	leftmost-innermost
$f(f(a, f(a, b)), g(f(a, b), g(b, a)))$	leftmost-outermost
$f(f(a, f(a, b)), g(f(a, b), g(b, a)))$	parallel-innermost

Needed Redexes

A redex r in M is needed if in any reduction to normal form from M either

1. some residual of r is reduced or
2. a redex that overlaps with a residual of r is reduced

- Intuition: a redex is needed if it is unavoidable
- In the case of OTRS only the first item above can hold

A needed strategy performs needed steps

Example

$$\begin{array}{llcl} \rho : & a & \longrightarrow & b \\ \vartheta : & f(x,b) & \longrightarrow & c \end{array}$$

a is needed in $f(a, a)$

$$f(a,a) \longrightarrow_{\rho} f(b,a) \longrightarrow_{\rho} f(b,b) \longrightarrow_{\vartheta} c$$

a is not needed in $f(a, a)$

$$f(a,a) \longrightarrow_{\rho} f(a,b) \longrightarrow_{\vartheta} c$$

Problem - Needed redexes need not exist

$$\begin{array}{llcl} \rho: & a & \to & b \\ \vartheta: & f(b,x) & \to & c \\ \theta: & f(x,b) & \to & c \end{array}$$

$f(a, a)$ has no needed redex

1. $f(a, a) \to_\rho f(a, b) \to_\theta c$
2. $f(a, a) \to_\rho f(b, a) \to_\vartheta c$

Problem - Needed redexes need not normalise

$$
\begin{array}{lcl}
a & \longrightarrow & b \\
f(x) & \longrightarrow & g(x,x) \\
g(a,b) & \longrightarrow & c \\
g(b,b) & \longrightarrow & g(b,b)
\end{array}
$$

$$
\begin{tikzcd}
f(a) \arrow[rr] \arrow[dd] & & f(b) \arrow[dd] \\
& g(b,a) \arrow[dr] & \\
g(a,a) \arrow[ur] \arrow[dr] & & g(b,b) \arrow[loop below] \\
& g(a,b) \arrow[ur] \arrow[d] & \\
& c &
\end{tikzcd}
$$

Why bother? - Normalisation Theorem I

Thm ([Huet,Lévy1991])

For orthogonal systems needed strategies normalise

Proof

Consider a standard normalising reduction sequence $d : M \twoheadrightarrow N$ and an infinite reduction sequence of needed steps from M: $M \to_{s_1} M_1 \to_{s_2} M_2 \to_{s_3} \ldots$.

$$\begin{tikzcd}
M \arrow[r, "s_1"'] \arrow[d, two heads, "d"'] & M_1 \arrow[r, "s_2"'] \arrow[dl, two heads, bend left, "d/s_1"'] & M_2 \arrow[r, "s_3"'] \arrow[dll, two heads, bend left, "d/(s_1;s_2)"] & M_3 \arrow[r, "s_4"'] \arrow[dlll, two heads, bend left, "d/(s_1;s_2;s_3)"] & \cdots \\
N & & & \cdots &
\end{tikzcd}$$

Each $d/(s_1;\ldots;s_i)$ is std and $|d| > |d/s_1| > |d/s_1;s_2| > \ldots$

On Deciding Neededness

- Every term in an OTRS has a needed redex
- However, it is not decidable whether a redex is needed or not

Consider the OTRS consisting of Combinatory Logic plus the rules

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

Consider determining whether any of the redexes s_1, s_2, s_3 are needed in $f(s_1, s_2, s_3)$
(Recall that the word problem for CL is undecidable)

On Deciding Neededness

- Nevertheless, for certain classes of OTRS some strategies can be proved needed
  1. Leftmost-outermost for left-normal (all function symbols occur to the left of variables). Eg. Combinatory Logic
  2. Leftmost-outermost for β

Needed Redexes and Standardization

- Recall

  A redex $r : M \rightarrow N$ is needed if it is unavoidable to perform r in order to reach a normal form
- Alternatively

  It is needed if one cannot get rid of it in any coinitial derivation

Q: How can one "get rid of r"?

A: Erase r from above

Needed Redexes and Standardization

Q: How can one "get rid of r"?

A: Erase r from above by

1. reducing an existing redex in M above r that has r in one of its erased arguments or
2. A derivation from M that creates a redex above r that has r in one of its erased the arguments

Consider the TRS $a \to b,\ f(x, b) \to c$

1. $f(b, b) \to c$
2. $f(b, a) \to f(b, b) \to c$

Note: These are standard derivations that erase b

Needed Redexes and Standardization

Consider the TRS $a \to b,\ f(x, b) \to c$

1. $f(b, b) \to c$
2. $f(b, a) \to f(b, b) \to c$

Note:

1. If we extend each derivation by prefixing it with an $r : f(a, b) \to f(b, b)$ we get a nonstandard derivation
2. Moreover, standardizing these extended derivations eliminates r

Consider the second item above:

$$f(a, a) \to f(b, a) \to f(b, b) \to c$$

Needed Redexes and Standardization

Consider the TRS $a \to b,\ f(x,b) \to c$

1. $f(b,b) \to c$
2. $f(b,a) \to f(b,b) \to c$

Note:

1. If we extend each derivation by prefixing it with an $r : f(a,b) \to f(b,b)$ we get a nonstandard derivation
2. Moreover, standardizing these extended derivations eliminates r

Consider the second item above:

$$\underline{f(a,a) \to f(b,a) \to f(b,b)} \to c$$

Reversible permutation

Needed Redexes and Standardization

Consider the TRS $a \to b,\ f(x,b) \to c$

1. $f(b,b) \to c$
2. $f(b,a) \to f(b,b) \to c$

Note:

1. If we extend each derivation by prefixing it with an $r : f(a,b) \to f(b,b)$ we get a nonstandard derivation
2. Moreover, standardizing these extended derivations eliminates r

Consider the second item above:

$$
\begin{array}{c}
f(a,a) \to f(b,a) \to f(b,b) \to c \\
f(a,a) \to f(a,b) \to f(b,b) \to c
\end{array}
$$

Needed Redexes and Standardization

Consider the TRS $a \to b,\ f(x,b) \to c$

1. $f(b,b) \to c$
2. $f(b,a) \to f(b,b) \to c$

Note:

1. If we extend each derivation by prefixing it with an $r : f(a,b) \to f(b,b)$ we get a nonstandard derivation
2. Moreover, standardizing these extended derivations eliminates r

Consider the second item above:

$$\begin{array}{l} f(a,a) \to f(b,a) \to f(b,b) \to c \\ f(a,a) \to \underline{f(a,b) \to f(b,b) \to c} \end{array}$$

Standardizing permutation

Needed Redexes and Standardization

Consider the TRS $a \to b,\ f(x, b) \to c$

1. $f(b, b) \to c$
2. $f(b, a) \to f(b, b) \to c$

Note:

1. If we extend each derivation by prefixing it with an $r : f(a, b) \to f(b, b)$ we get a nonstandard derivation
2. Moreover, standardizing these extended derivations eliminates r

Consider the second item above:

$$f(a, a) \to f(b, a) \to f(b, b) \to c$$
$$f(a, a) \to f(a, b) \to f(b, b) \to c$$
$$f(a, a) \to f(a, b) \to c$$

Neededness - Alternative Definition

A redex $r : M \to N$ is needed iff

$$\forall P\ \forall e : N \twoheadrightarrow P,\ |\mathrm{std}(r; e)| > |\mathrm{std}(e)|$$

- This definition coincides with the previous one
- Its appeal: allows generalization to needed derivations

Needed Derivations

We generalize our previous notion of needed redexes to needed derivations

A derivation $d : M \twoheadrightarrow N$ is needed if

$$\forall P\ \forall e : N \twoheadrightarrow P,\ |\mathrm{std}(d;e)| > |\mathrm{std}(e)|$$

A needed strategy is a (multi-step) strategy $\mathbb{F}$ s.t. $\forall M$

$$M \to^{+} \mathbb{F}(M) \text{ is a needed derivation}$$

Needed Derivations

In Non-Orthogonal TRS needed redexes may not exist (as already seen)
But needed derivations always do!

Prop.

Every standard, normalising derivation is needed

Proof

Immediate from definition

Example

$$\begin{array}{lcl} a & \longrightarrow & b \\ f(b,x) & \longrightarrow & g(c) \\ f(x,b) & \longrightarrow & g(c) \\ g(c) & \longrightarrow & d \end{array}$$

Although a in $f(a, a)$ is not needed, it extends to a needed derivation

$$d : f(a, a) \rightarrow f(a, b) \rightarrow g(c)$$

External Redexes

One way of constructing needed derivations is by contracting external redexes

A redex is external to a coinitial derivation if its residuals are not nested by other redexes in the course of the derivation. A redex is external if it is external to any derivation.

$$\begin{array}{lcl} a & \longrightarrow & b \\ f(x,b) & \longrightarrow & g(a) \end{array}$$

a is not external, a is in the term $f(a,a)$

Note: External redexes are needed (the converse does not hold)

Finite Normalisation Cones

Idea:

1. Suppose there are only a finite number of different normalising derivations from M modulo Lévy permutation equivalence
2. Measure M by the longest such one
3. Show that needed derivations decrease this measure

Finite Normalisation Cones

A normalisation cone from M is a set $\{e_i^M : M \twoheadrightarrow P_i\}$ of normalising derivations s.t. for each normalising derivation $f : M \twoheadrightarrow N$, there exists a unique i, $f \equiv e_i$.

A TRS enjoys finite normalisation cones (FNC) when for any M there exists a finite normalisation cone for M.

Finite Normalisation Cone

Example

All OTRS: normalising derivations are unique modulo $\equiv$ in that setting

Non-Example

Consider the TRS

$$\begin{array}{rcl} a & \rightarrow & b \\ a & \rightarrow & a \end{array}$$

and the derivations

$$a \rightarrow b$$
$$a \rightarrow a \rightarrow b$$
$$a \rightarrow a \rightarrow a \rightarrow b$$
$$\ldots$$

Normalisation Theorem II

Thm

Needed strategies normalise for TRS enjoying finite normalisation cones

Proof

1. Define depth(M) to be the longest derivation in the finite normalisation cone of M: $\{e_i^M : M \twoheadrightarrow P_i\}$ (each e_i^M may be assumed standard)
2. Show that if $d : M \twoheadrightarrow N$ is a needed derivation, then depth(M) $>$ depth(N)

$$\begin{tikzcd} M \arrow[d, twoheadrightarrow, "d"'] \arrow[dr, twoheadrightarrow, "e_{j*}^M"] \arrow[dr, phantom, "\equiv" near start, shift right=3ex] & \\ N \arrow[r, twoheadrightarrow, "e_j^N"'] & P_j \end{tikzcd}$$

$$|e_{j*}^M| = |\mathrm{std}(d; e_j^N)| > |e_j^N|$$

How do we use this result?

- Find classes of TRS that enjoy FNC
- As mentioned, all OTRS do (normalising cones are not only finite, they are singletons)
- But, what about non-orthogonal TRS?

How do we use this result?

- Weakly OTRS (admit trivial critical pairs)? No (van Oostrom)

$$\begin{array}{lcl} a & \rightarrow & f(a) \\ f(b) & \rightarrow & b \\ f(x) & \rightarrow & b \end{array}$$

- Even though FNC fails already for weakly OTRS, the story is different for calculi with explicit substitutions
- Next talk: We'll spell out the details
- Problem: Characterize interesting classes of TRS that satisfy FNC

Credits

- Neededness and Normalisation Theorem I: [Huet,Lévy1991]
- Normalisation Theorem II (i.e. extension to non-orthogonal case): [Melliès1996,2000]
  - He developed the results in an axiomatic rewriting framework and in terms of 2-categorical models of rewriting
  - This framework is syntax free (i.e. independent of the structure of rewritten objects)
  - Many rewriting formats are thus captured