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
Consider the TRS

\[ \rho : \quad a \rightarrow b \]
\[ \vartheta : \quad f(x, a) \rightarrow g(x, x) \]

and the term

\[ f(a, a) \]

It has three redexes, \( r \), \( s \) and \( t \):

\[ f(a, a) \quad f(a, a) \quad f(a, a) \]
Example

\[ \rho : \ a \rightarrow b \]
\[ \vartheta : \ f(x, a) \rightarrow g(x, x) \]

Consider the redexes \( r \) and \( s \):

\[ f(a, a) \quad f(a, a) \]

Reducing \( s \) leaves a “leftover” or residual of \( r \)

\[ f(a, a) \rightarrow_{\rho} f(b, a) \]

Likewise reducing \( r \) leaves a “leftover” or residual (two actually) of \( s \)

\[ f(a, a) \rightarrow_{\vartheta} g(a, a) \]

Note: \( r \) and \( s \) do not overlap
Example

\[ \rho : \ a \rightarrow b \]
\[ \vartheta : \ f(x, a) \rightarrow g(x, x) \]

Consider the redexes \( r \) and \( t \):

\[ f(a, a) \]
\[ f(a, a) \]

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 \textit{erased}. Eg. replacing \( \vartheta \) by \( f(x, a) \rightarrow b \), note \( a \) is erased below

\[ f(a, a) \rightarrow_{\vartheta} b \]
Definition of Residuals

Assume \( \rho \)-redex \( r \) and \( \vartheta \)-redex \( s \) in \( M \) and \( M \rightarrow_r N \)

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

\[ \rho : \quad a \quad \rightarrow \quad b \]
\[ \vartheta : \quad f(x, b) \quad \rightarrow \quad g(x, x) \]

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

\[ f(a, a) \rightarrow_{\rho} f(a, b) \]
1 Residuals
   • Examples and Definition
   • Equivalence of Derivations

2 Standardization

3 Needed Strategies
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,

\[
\begin{array}{c}
M_1 \xrightarrow{r} M_2 \xrightarrow{d} \ldots \xrightarrow{d} M_n \\
s \downarrow \\
v \downarrow \\
u \downarrow 
\end{array}
\]
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. $d : M \rightarrow N$
2. $V$ is the set of residuals of elements of $U$ after $d$

$$V = \{ v \mid \exists u \in U, \ u/dv \}$$
A derivation
\[ d : M = M_1 \rightarrow_{r_1} M_2 \rightarrow_{r_2} M_3 \ldots \rightarrow_{r_{n-1}} M_n \rightarrow_{r_n} M_{n+1} \]
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 \( d : M \rightarrow M_{n+1} \) 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 and the multi-redex
\[ \rho : \ a \rightarrow b \]
\[ \vartheta : \ f(x, b) \rightarrow g(x, x) \]
\[ \langle f(a, b), \{r, s\} \rangle \]

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

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

\[
M_1 \rightarrow_{r_1} M_2 \rightarrow_{r_2} M_3 \rightarrow_{r_3} \ldots
\]

s.t. for each \( i \) the derivation

\[
M_1 \rightarrow_{r_1} M_2 \rightarrow_{r_2} M_3 \rightarrow_{r_3} \ldots M_{i-1} \rightarrow_{r_{i-1}} M_i
\]
develops \( \langle M, U \rangle \) partially

Informally,

Contraction (only) of residuals of a fixed set \( U \) of redexes in \( M \) eventually terminates
Lemma (Parallel Moves)

For every two coinitial, non-overlapping redexes $r : M \rightarrow P$ and $s : M \rightarrow Q$ 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

\[
\begin{array}{c}
M \xrightarrow{r} P \\
\downarrow s & \downarrow d_s \\
Q \xrightarrow{d_r} N
\end{array}
\]
Basic tile and Equivalence of Derivations

- Basic tile provides a convenient mechanism for defining a notion of equivalence of derivations.

- **Intuition:** \( d : M \rightarrow N \) and \( e : M \rightarrow N \) are “equivalent” if they do the same “work” but in different “order”.

\[
\begin{align*}
  f(a, a) & \xrightarrow{a} f(b, a) \\
  \downarrow a & \quad \downarrow a \\
  f(a, b) & \xrightarrow{a} f(b, b) \\
  f(\Box, b) & \quad f(\Box, b) \\
  \downarrow & \quad \downarrow \\
  g(a, a) & \xrightarrow{a} g(a, b) \xrightarrow{a} g(b, b)
\end{align*}
\]
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.

\[
\begin{array}{cccc}
R & \xrightarrow{f_1} & M & \xrightarrow{r} \to P \\
\downarrow{s} & & \downarrow{d_s} & \\
Q & \xrightarrow{d_r} & N & \xrightarrow{f_2} \to S
\end{array}
\]

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 connecting \( f \) and \( g \).
Lévy Permutation Equivalence - Example Revisited

\[
\begin{align*}
&f(a, a) \rightarrow f(b, a) \\
&f(a, b) \rightarrow f(b, b) \\
&f(\square, b) \rightarrow f(\square, b) \\
&g(a, a) \rightarrow g(a, b) \rightarrow g(b, b)
\end{align*}
\]
Epimorphism

Lemma ([Berry] for $\lambda$)

\[ d; e \equiv d; f \text{ implies } e \equiv f \]

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 \( \lambda \))**

Let \( d : M \to P \) and \( e : M \to Q \) be coinitial in an OTRS. Then:

\[
M \xrightarrow{d} P \quad \equiv \quad e' \quad \equiv \quad e''
\]

\[
Q \xrightarrow{d'} N \quad \equiv \quad e''
\]

\[
N \xrightarrow{f} O
\]

The category generated by the reduction graph of an OTRS with \( \equiv \) as identity on arrows enjoys pushouts
1 Residuals

2 Standardization

3 Needed Strategies
The idea: for any derivation from $M$ to $N$, there is a canonical derivation from $M$ to $N$ that computes redexes “outside-in”

\[
\begin{align*}
a & \rightarrow b \\
f(x, b) & \rightarrow g(x, x)
\end{align*}
\]

Not standard

\[
f(a, b) \rightarrow f(b, b) \rightarrow g(b, b)
\]

Standard

\[
f(a, b) \rightarrow g(a, a) \rightarrow g(b, a) \rightarrow g(b, b)
\]
Examples - Uniqueness

\[ a \rightarrow b \]
\[ f(x, b) \rightarrow g(x, x) \]

Both these derivations are standard

\[ f(a, a) \rightarrow f(b, a) \rightarrow f(b, b) \]
\[ f(a, a) \rightarrow f(a, b) \rightarrow f(b, b) \]

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

![Diagram](diag.png)

**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 $g$
Example 1

\[ f(a, b) \xrightarrow{a} f(b, b) \]

\[ f(\Box, b) \xrightarrow{} \]

\[ g(b, b) \]
Example 1

\[
f(a, b) \xrightarrow{a} f(b, b) \\
\]

\[
f(\Box, b) \xrightarrow{\Box} f(\Box, b) \\
\]

\[
g(a, a) \xrightarrow{a} g(b, b) \\
\]
Defining Standard Derivations Through Permutation

Example 1

\[ f(a, b) \xrightarrow{a} f(b, b) \]
\[ f(\square, b) \]
\[ g(a, a) \xrightarrow{a} g(b, b) \]

Example 2

\[ f(a, a) \xrightarrow{a} f(b, a) \]
\[ a \]
\[ f(b, b) \]
\[ f(\square, b) \]
\[ g(b, b) \]
Defining Standard Derivations Through Permutation

**Example 1**

\[
\begin{align*}
  f(a, b) & \xrightarrow{a} f(b, b) \\
  f(\Box, b) & \quad f(\Box, b) \\
  g(a, a) & \xrightarrow{a} g(b, b)
\end{align*}
\]

**Example 2**

\[
\begin{align*}
  f(a, a) & \xrightarrow{a} f(b, a) \\
  a & \quad \sim a \\
  f(a, b) & \xrightarrow{a} f(b, b) \\
  f(\Box, b) & \quad f(\Box, b) \\
  g(b, b)
\end{align*}
\]
Defining Standard Derivations Through Permutation

Example 1

\[ f(a, b) \xrightarrow{a} f(b, b) \]
\[ f(\Box, b) \]
\[ g(a, a) \xrightarrow{a} g(b, b) \]

Example 2

\[ f(a, a) \xrightarrow{a} f(b, a) \]
\[ f(\Box, b) \]
\[ g(a, a) \xrightarrow{a} g(b, b) \]
Standardizing Permutation

We need one more ingredient for defining standard derivations

There is a **standardizing** permutation from \( f : M \rightarrow N \) to \( g : M \rightarrow N \) (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

\[
\begin{array}{ccccccc}
R & \xrightarrow{f_1} & M & \xrightarrow{r} & P \\
\downarrow s & & \downarrow d_s & & \downarrow d_r \\
Q & \xrightarrow{d_r} & N & \xrightarrow{f_2} & S \\
\end{array}
\]

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 $f : M \rightarrow N$ to $g : M \rightarrow N$ (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

```
R \xrightarrow{f_1} M \xrightarrow{r} P
\downarrow s \quad \downarrow d_s
Q \xrightarrow{d_r} N \xrightarrow{f_2} S
```

In fact, since $s$ nests $r$, $d_s$ will consist of just one redex
So we can write this definition more accurately as follows
There is a standardizing permutation from \( f : M \rightarrow N \) to \( g : M \rightarrow N \) (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

\[
\begin{array}{cccc}
R & \xrightarrow{f_1} & M & \xrightarrow{r} & P \\
\downarrow{s} & & \downarrow{s'} & & \\
Q & \xrightarrow{d_r} & N & \xrightarrow{f_2} & S \\
\end{array}
\]
A derivation $f$ is \textbf{standard} if there is no derivation $g$ s.t.

$$[f]_\sim \Rightarrow [g]_\sim$$

Note: $[f]_\sim$ 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

\[
\begin{align*}
a & \rightarrow b \\
f(x, b) & \rightarrow g(x, x)
\end{align*}
\]

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

\[
\begin{array}{c}
f(a, b) \xrightarrow{a} f(b, b) \\
f(\square, b) \downarrow & f(\square, b) \downarrow \\
g(a, a) \xrightarrow{a} g(b, b)
\end{array}
\]

Indeed,

\[
d \Rightarrow f(a, b) \rightarrow g(a, a) \rightarrow g(b, a) \rightarrow g(b, b)
\]
Example 2 - Revisited

\[ a \rightarrow b \]
\[ f(x, b) \rightarrow g(x, x) \]

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

Indeed, \( d \simeq f(a, a) \rightarrow f(a, b) \rightarrow f(b, b) \rightarrow g(b, b) \)

\( \Rightarrow f(a, a) \rightarrow f(a, b) \rightarrow g(a, a) \rightarrow g(b, a) \rightarrow g(b, b) \)
Standardization Theorem

Thm

1 Existence
For every $d : M \rightarrow N$, there exists a standard derivation $\text{std}(d) : M \rightarrow N$ and $d_1, \ldots, d_n$ s.t.

$$[d] \Rightarrow [d_1] \Rightarrow \ldots \Rightarrow [d_n] \Rightarrow [\text{std}(d)]$$

2 Uniqueness
Let $d, e : M \rightarrow N$ s.t. $d \equiv e$ (i.e. $d$ and $e$ are Lévy permutation equivalent). Then

$$[\text{std}(d)] = [\text{std}(e)]$$
Computing Standard Derivations

1. Repeatedly apply standardizing permutations on \( \sim \)-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 \( d : M \to N \) compute \( \text{std}(d) \) 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")]

1 Residuals

2 Standardization

3 Needed Strategies
   - Needed Redexes
   - Needed Redexes and Standardization
   - Neededness for Non-Orthogonal Systems
A (one-step or many step) reduction strategy for a TRS $\mathcal{R}$ is a function $\text{IF} : \mathcal{T}(\Sigma) \rightarrow \mathcal{T}(\Sigma)$ s.t.

1. $\text{IF}(M) = M$, if $M$ is in $\mathcal{R}$-normal form
2. $M \rightarrow^+ \text{IF}(M)$, otherwise

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

$$M \rightarrow^+ \text{IF}(M) \rightarrow^+ \text{IF}(\text{IF}(M)) \rightarrow^+ \text{IF}(\text{IF}(\text{IF}(M))) \rightarrow^+ \ldots$$
Example

Consider the TRS

\[
\begin{align*}
  f(a, x) & \rightarrow x \\
  f(b, x) & \rightarrow b \\
  g(a, x) & \rightarrow a \\
  g(b, x) & \rightarrow x
\end{align*}
\]

\[
\begin{align*}
  f(f(a, f(a, b)), g(f(a, b), g(b, a))) & \text{ leftmost-innermost} \\
  f(f(a, f(a, b)), g(f(a, b), g(b, a))) & \text{ leftmost-outermost} \\
  f(f(a, f(a, b)), g(f(a, b), g(b, a))) & \text{ parallel-innermost}
\end{align*}
\]
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

\[ \rho : \ a \rightarrow b \]
\[ \varnothing : \ f(x, b) \rightarrow c \]

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

\[ f(a, a) \rightarrow_\rho f(b, a) \rightarrow_\rho f(b, b) \rightarrow_\varnothing c \]

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

\[ f(a, a) \rightarrow_\rho f(a, b) \rightarrow_\varnothing c \]
Problem - Needed redexes need not exist

\[ \rho : \quad a \rightarrow b \]
\[ \vartheta : \quad f(b, x) \rightarrow c \]
\[ \theta : \quad f(x, b) \rightarrow c \]

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

1. \( f(a, a) \xrightarrow{\rho} f(a, b) \xrightarrow{\theta} c \)
2. \( f(a, a) \xrightarrow{\rho} f(b, a) \xrightarrow{\vartheta} c \)
Problem - Needed redexes need not normalise

\[
\begin{align*}
a & \rightarrow b \\
f(x) & \rightarrow g(x, x) \\
g(a, b) & \rightarrow c \\
g(b, b) & \rightarrow g(b, b)
\end{align*}
\]
Why bother? - Normalisation Theorem I

Thm ([Huet, Lévy 1991])

For orthogonal systems needed strategies normalise

Proof

Consider a standard normalising reduction sequence \( d : M \rightarrow N \) and an infinite reduction sequence of needed steps from \( M \):

\[
\begin{align*}
M \rightarrow_{s_1} M_1 \rightarrow_{s_2} M_2 \rightarrow_{s_3} M_3 \rightarrow_{s_4} \cdots
\end{align*}
\]

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

\[
\begin{align*}
g(a, b, x) & \rightarrow c \\
g(x, a, b) & \rightarrow c \\
g(b, x, a) & \rightarrow c
\end{align*}
\]

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 $\beta$
1 Residuals

2 Standardization

3 Needed Strategies
   - Needed Redexes
   - Needed Redexes and Standardization
   - Neededness for Non-Orthogonal Systems
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.
**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 $a \rightarrow b$, $f(x, b) \rightarrow c$

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

Note: These are standard derivations that erase $b$
Needed Redexes and Standardization

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

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

Note:

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

Consider the second item above:

$$f(a, a) \rightarrow f(b, a) \rightarrow f(b, b) \rightarrow c$$
Needed Redexes and Standardization

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

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

Note:

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

Consider the second item above:

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

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:

\[
\begin{align*}
  f(a, a) & \to f(b, a) \to f(b, b) \to c \\
  f(a, a) & \to f(a, b) \to f(b, b) \to c
\end{align*}
\]
Needed Redexes and Standardization

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

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

Note:

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

Consider the second item above:

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

Standardizing permutation
Needed Redexes and Standardization

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

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

Note:

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

Consider the second item above:

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

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

$$f(a, a) \rightarrow f(a, b) \rightarrow c$$
A redex $r : M \to N$ is \textit{needed} iff

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

- This definition coincides with the previous one
- Its appeal: allows generalization to needed \textit{derivations}
1. Residuals

2. Standardization

3. Needed Strategies
   - Needed Redexes
   - Needed Redexes and Standardization
   - Neededness for Non-Orthogonal Systems
Needed Derivations

We generalize our previous notion of needed redexes to needed derivations.

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

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

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

$$M \rightarrow^+ \ \mathcal{F}(M) \text{ is a needed derivation}$$
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

\[
\begin{align*}
    a & \rightarrow b \\
    f(b, x) & \rightarrow g(c) \\
    f(x, b) & \rightarrow g(c) \\
    g(c) & \rightarrow d
\end{align*}
\]

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.

\[ a \rightarrow b \]
\[ f(x, b) \rightarrow g(a) \]

\( a \) 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
A normalisation cone from $M$ is a set $\{e_i^M : M \to P_i\}$ of normalising derivations s.t. for each normalising derivation $f : M \to N$, 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

\[
\begin{align*}
a & \rightarrow b \\
& a & \rightarrow a \\
\end{align*}
\]

and the derivations

\[
\begin{align*}
a & \rightarrow b \\
& a \rightarrow a \rightarrow b \\
& a \rightarrow a \rightarrow a \rightarrow b \\
& \ldots
\end{align*}
\]
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^M_i : M \to P_i\} (each $e^M_i$ may be assumed standard)

2. Show that if $d : M \to N$ is a needed derivation, then depth($M$) $>$ depth($N$)

\[
\begin{align*}
M & \xrightarrow{d} e^M_{j*} \\
N & \xrightarrow{e^N_j} P_j \\
|e^M_{j*}| & = |\text{std}(d; e^N_j)| > |e^N_j|
\end{align*}
\]
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)

\[
\begin{align*}
  a & \rightarrow f(a) \\
  f(b) & \rightarrow b \\
  f(x) & \rightarrow b
\end{align*}
\]

- 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, Lévy 1991]
- Normalisation Theorem II (i.e. extension to non-orthogonal case): [Melliès 1996, 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