Termination

Termination is essential to proof correctness of programs.

But

Termination is an undecidable property.

Undecidability of termination

Let $a_1, a_2, a_3, \ldots$ be an enumeration of all the algorithms on integers. We define the following functions:

$\mathtt{end}(i, n) \equiv 1$ if $a_i(n)$ terminates
$\mathtt{end}(i, n) \equiv 0$ if $a_i(n)$ $\neg$ terminates

$\mathtt{Diag}(i) \equiv$ if $\mathtt{end}(i, i) = 1$ then loop
else stop

For every i, $\mathtt{Diag}(i)$ terminates iff $a_i(i)$ does not terminate.
But $\mathtt{Diag}$ is an algorithm, so that $\exists a_j$ s.t. $\mathtt{Diag} = a_j$. We then have $\mathtt{Diag}(j)$ terminates iff $a_j(j)$ terminates, that is

$$a_j(j) \text{ terminates iff } a_j(j) \text{ does not terminate.}$$

Which is the error in the proof? The existence of the function $\mathtt{end}$.

Termination is not trivial

Termination of a very simple system like

$$\mathcal{R} = \{f(g(x), y) \mapsto f(y, y)\}$$

is not even trivial!

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

Techniques to show termination

- Reduction orders
  - Particular case: interpretations
  - Example of interpretation: polynomial orders
- Useful orders:
  - Lexicographic order
  - Multi-set order
- Simplification orders
  - General result
  - Example : RPO
- Combination of orders:
  - Motivations
  - Postponement
  - Projection/simulation
- Dependency pairs

Recall

The symbol $f \in \Sigma$ is monotonic w.r.t the relation R iff
$a_i \; R \; b_i$ implies $f(a_1, \ldots, a_i, \ldots, a_n) \; R \; f(a_1, \ldots, b_i, \ldots, a_n)$.

Alternatively, a relation R over $\mathcal{T}(\mathcal{X}, \Sigma)$ is stable by context iff
$t \; R \; t'$ implies $u[t]_p \; R \; u[t']_p$ for every term u and position $p \in Pos(u)$.

A relation R over $\mathcal{T}(\mathcal{X}, \Sigma)$ is stable by substitution iff
$t \; R \; t'$ implies $\theta(t) \; R \; \theta(t')$ for every substitution θ.

A relation R is well-founded (WF) is there is no infinite sequence of the form
$a_1 \; \mathcal{R} \; a_2 \; \mathcal{R} \; a_3 \; \mathcal{R} \; \ldots$

Termination by reduction orders

Pre-order: reflexive and transitive relation.
Partial order: reflexive, antisymmetric and transitive relation.
Strict order: irreflexive and transitive (and thus antisymmetric) relation.

A strict order $\succ$ over a signature Σ is a reduction order iff

1. $\succ$ is stable by context
2. $\succ$ is stable by substitution
3. $\succ$ is well-founded

Why reduction orders are important?

Theorem (Termination and Reduction Order)

A rewriting system $\mathcal{R}$ terminates iff there exists a reduction order $\succ$ such that $l \succ r$ for every rewriting rule $l \mapsto r \in \mathcal{R}$.

Proof.

Let $\mathcal{R}$ be a terminating rewriting system. The relation $\rightarrow^{+}_{\mathcal{R}}$ holds for every rule $l \mapsto r \in \mathcal{R}$. Moreover, $\rightarrow^{+}_{\mathcal{R}}$ is a reduction order (stable by substitution, stable by context and WF). Therefore the property holds for $\succ := \rightarrow^{+}_{\mathcal{R}}$.

Let $\succ$ be a reduction order s.t. $l \succ r$ holds for every rule $l \mapsto r \in \mathcal{R}$.
Consider any step $s \rightarrow_{\mathcal{R}} v$. By definition $s = u[\sigma(l)]_p$ and $v = [\sigma(r)]_p$, for some rule $l \mapsto r \in \mathcal{R}$, some substitution σ, some term u, and some position p of u. Therefore $l \succ r$ holds by hypothesis. Since $\succ$ is a reduction order, then also $s \succ v$ holds. Therefore every step $s \rightarrow_{\mathcal{R}} v$ generates a pair $s \succ v$. But $\succ$ is WF by hypothesis, so that the relation $\rightarrow_{\mathcal{R}}$ must be also terminating. □

How does it work?

Does $\mathcal{R}$ terminate?

$$\Sigma = \{\mathtt{t}/0, \mathtt{por}/2\} \qquad \mathcal{R} = \left\{ \begin{array}{lcl} \mathtt{por}(x, \mathtt{t}) & \mapsto & \mathtt{t} \\ \mathtt{por}(\mathtt{t}, x) & \mapsto & \mathtt{t} \end{array} \right\}$$

The number of symbol decreases.... Let us try to construct a reduction order!

- Let $|v|$ be the size of the term v. Consider the following order "$s \succ_1 t$ iff $|s| > |t|$".
  - We have $\mathtt{por}(x, \mathtt{t}) \succ_1 \mathtt{t}$ and $\mathtt{por}(\mathtt{t}, x) \succ_1 \mathtt{t}$.
  - But $\succ_1$ is not a reduction order since $\succ_1$ is not stable by substitution: $\mathtt{por}(x, \mathtt{por}(y, \mathtt{t})) \succ_1 \mathtt{por}(y, y)$ but $\mathtt{por}(\mathtt{t}, \mathtt{por}(\mathtt{por}(\mathtt{t}, \mathtt{t}), \mathtt{t})) \not\succ_1 \mathtt{por}(\mathtt{por}(\mathtt{t}, \mathtt{t}), \mathtt{por}(\mathtt{t}, \mathtt{t}))$.
  - We conclude that $\succ_1$ is not a reduction order.
- Let $|v|_x$ be the number of free occurrences of the variable x in v. Consider the following order "$s \succ_2 t$ iff $|s| > |t|$ and for every variable x $|s|_x \geq |t|_x$".
  - We have $\mathtt{por}(x, \mathtt{t}) \succ_2 \mathtt{t}$ and $\mathtt{por}(\mathtt{t}, x) \succ_2 \mathtt{t}$.
  - And we can prove that $\succ_2$ is a reduction order (exercice).

Then by applying the previous theorem, we have proved that $\mathcal{R}$ is terminating.

Interpretation as particular case of reduction order

A reduction order can also be defined on the interpretation of terms, and not directly on the terms themselves.

Definition

Let $>_{\mathcal{A}}$ be a WF strict order over the domain of the Σ-algebra $\mathcal{A}$. The associated order $\succ$ over the corresponding set of terms is given by:

$$s \succ t \text{ iff } \Phi(s) >_{\mathcal{A}} \Phi(t) \text{ for all homomorphisms } \Phi : \mathcal{T}(\mathcal{X}, \Sigma) \to \mathcal{A}$$

- This order on terms is then based on the order relating the interpretations of s and t.
- The definition considers all possible valuations of variables in the domain of $\mathcal{A}$.

Theorem

Let $\mathcal{A}$ be a Σ-algebra equipped with a WF strict order $>_{\mathcal{A}}$ over the domain of $\mathcal{A}$. If for every $f \in \Sigma$, the interpretation $f^{\mathcal{A}}$ is **monotonic** *w.r.t. $>_{\mathcal{A}}$, then $\succ$ is a reduction order.*

Example: polynomial orders

A polynomial Σ-algebra $\mathcal{P}_{\mathbb{N}}$ is defined by:

- A domain **D** which is a subset of $\mathbb{N}^+$, *i.e.* $\mathbf{D} \subseteq \mathbb{N}^+$.
- A polynomial P_f (with n indeterminates and coefficients in $\mathbb{N}$) for each $f/n \in \Sigma$, such that $f^{\mathcal{P}_{\mathbb{N}}}(a_1, \ldots, a_n) = P_f(a_1, \ldots, a_n)$.

Example

Let $\Sigma = \{a/0, f/2, g/2\}$. Consider a polynomial Σ-algebra with domain $\mathbf{D} = \{n \in \mathbb{N} \mid n \geq 2\} \subseteq \mathbb{N}^+$ and polynomial interpretations $P_a = 2$, $P_f(n, m) = n.m$ and $P_g(n, m) = 3.n + m + 1$. Consider a valuation σ on variables such that $\sigma(x) = 5$. Then we have $\sigma(f(a, g(a, x))) = 2.(3.2 + 5 + 1) = 24$.

Problem

Polynomials are not necessarily monotonic, for example if $P_f(x, y) = x^2$ we have $3 > 2$ but $P_f(2, 3) = 4 \not> 4 = P_f(2, 2)$.

Towards a polynomial order as interpretation

A polynomial P is completely monotonic iff P depends on all its indeterminates.

Example

$P(x, y) = 3.x + y + 2$ and $P(x, y) = x.y$ are both completely monotonic but $P(x, y) = x + 2$ is not completely monotonic.

Theorem

Let $\mathcal{P}_{\mathbb{N}}$ be a polynomial Σ-algebra. If every $f^{\mathcal{P}_{\mathbb{N}}}$ is a **completely monotonic** *polynomial, then the order $\succ$ associated to $\succ_{\mathcal{P}_{\mathbb{N}}}$ is a reduction order.*

How does it work?

Does $\mathcal{R}$ terminate?

$$\Sigma = \{f/2, g/2\} \qquad \mathcal{R} = \left\{ \; f(x, g(y, z)) \quad \mapsto \quad g(f(x, y), f(x, z)) \; \right\}$$

1. Define a polynomial for every function symbol in the signature by respecting the arities: in our case $P_f(x, y) = x.y$ et $P_g(x, y) = 2.x + y + 1$.
2. For each rule $l \mapsto r \in \mathcal{R}$, prove that $l \succ r$. In our case, we need to prove that $f(x, g(y, z)) \succ g(f(x, y), f(x, z))$: this is equivalent to prove that $\sigma(x).(2.\sigma(y) + \sigma(z) + 1) \succ_{\mathcal{P}_{\mathbb{N}}} 2.\sigma(x).\sigma(y) + \sigma(x).\sigma(z) + 1$ holds for every σ mapping x, y, z to the domain to be defined.
3. Define the domain as the one in which all the inequalities are valid: in our case $\mathbf{D} = \mathbb{N} - \{0, 1\} \subseteq \mathbb{N}^{+}$.

Lexicographic order - particular case

Let $(A_1, >_{A_1})$ and $(A_2, >_{A_2})$ be two strict ordered sets.

$$(x, y) >_{lex} (x', y') \text{ iff } (x >_{A_1} x') \text{ or } (x = x' \text{ and } y >_{A_2} y')$$

Example

$$(4, \text{"}abc\text{"}) >_{lex} (3, \text{"}abc\text{"}) >_{lex} (2, \text{"}abcde\text{"}) >_{lex}$$
$$(2, \text{"}bcde\text{"}) >_{lex} (2, \text{"}e\text{"}) >_{lex} (1, \text{"}e\text{"}) >_{lex} (0, \epsilon)$$

Lexicographic order - General case

If every $>_{A_i}$ is a strict order over the set $\mathcal{A}_i$, then $>_{lex}$ is a strict order over $\mathcal{A}_1 \times \ldots \times \mathcal{A}_n$ defined as follows:

$$(x_1, \ldots, x_n) >_{lex} (x'_1, \ldots, x'_n) \text{ iff } \quad \exists 1 \leq j \leq n \; (x_j >_{A_j} x'_j \text{ and } \forall 1 \leq i < j \;\; x_i = x'_i)$$

Theorem

Every order $>_{A_i}$ over $\mathcal{A}_i$ is well-founded iff the lexicographic order $>_{lex}$ over $\mathcal{A}_1 \times \ldots \times \mathcal{A}_n$ is well-founded.

How does it work?

Does the following program terminate?

ackerman(0,n)	$\mapsto$	n+1
ackerman(m+1,0)	$\mapsto$	ackerman(m,1)
ackerman(m+1,n+1)	$\mapsto$	ackerman(m,ackerman(m+1,n))

Proof.

We show that ackerman(m, n) terminates by induction on (m, n) w.r.t. the lexicographic order. $\square$

Another example

Does the following program terminate?

$$\Sigma = \{f/1, g/1\} \qquad \mathcal{R} = \left\{ \begin{array}{lcl} f(f(x)) & \mapsto & g(f(x)) \\ g(g(x)) & \mapsto & f(x) \end{array} \right\}$$

Proof.

- Define the order $t > u$ iff $(|t|, |t|_f) >_{lex} (|u|, |u|_f)$.
- Show that $>$ is a reduction order.
- Show that $f(f(x)) > g(f(x))$ and $g(g(x)) > f(x)$.

□

Multi-set order

A multi-set over a set $\mathcal{A}$ is a function $\mathcal{M} : \mathcal{A} \to \mathbb{N}$. It is finite if $\mathcal{M}(x) > 0$ only for a finite number of elements of $\mathcal{A}$.

Example

The multiset $\{a, a, b\}$ is represented by the function $\mathcal{M}$ on the set $\{a, b\}$ such that $\mathcal{M}(a) = 2$ and $\mathcal{M}(b) = 1$.

Let $\mathcal{M}$ and $\mathcal{N}$ be two multi-sets. The multi-set union is defined by $(\mathcal{M} \uplus \mathcal{N})(a) = \mathcal{M}(a) + \mathcal{N}(a)$.

Example

$\{a, a, b\} \uplus \{a, a, b, b\} = \{a, a, a, a, b, b, b\}$.

Multi-set order

Let $\succ$ a strict order. The associated relation $\succ_{mul}$ is given by the transitive closure of the following relation $\succ_{mul}$:

$$\mathcal{M} \uplus \{x\} \succ_{mul} \mathcal{M} \uplus \{y_1, \ldots, y_n\}, \text{ where } n \geq 0 \text{ and } \forall i, x \succ y_i.$$

Example

$\{5, 3, 1, 1\} \succ_{mul} \{4, 3, 3, 1\}$.
Since $\{5, 3, 1, 1\} \succ_{mul} \{4, 3, 3, 1, 1\} \succ_{mul} \{4, 3, 3, 1\}$

Exercise : If $\succ$ is a strict oder, then $\succ_{mul}$ is a strict order.

Theorem

Let $\succ$ be a strict order over $\mathcal{A}$, then $\succ$ is WF iff $\succ_{mul}$ is WF.

How does it work?

A rich but bored man decides to have fun every day with his money (in euros) in the following way:

- either he throw a coin in the fountain,
- or he changes a banknote into a finite number of coins of any amount.

Show that the man necessarily becomes poor.

Proof.

- Represent the initial amount of money by a multi-set, where the elements are order as follows.

$$b(500) > b(200) > b(100) > b(50) > b(20) > b(10) > b(5) >$$
$$c(2) > c(1) > c(.50) > c(.20) > c(.10) > c(.05) > c(.02) > c(.01)$$

- Represent the daily activity of the man by a decreasing order on multi-sets.

□

Other known examples

- Hercules defeats Hydra
- Cut elimination in Gentzen style systems
- Amoebae reproduction
- Recursive Path Orderings

Simplification orders

A simplification order over $\mathcal{T}(\mathcal{X},\Sigma)$ is an order $\succ$ such that

1. $\succ$ is stable by context
2. $\succ$ is stable by substitution
3. $t \triangleright u$ implies $t \succ u$

Example: embedding

The relation $s \trianglerighteq_{emb} t$ holds iff one of the following cases hold

- s and t are the same variable
- $s = f(s_1, \ldots, s_n)$ and $t = f(t_1, \ldots, t_n)$ and $\forall i \; s_i \trianglerighteq_{emb} t_i$
- $s = f(s_1, \ldots, s_n)$ and there is j s.t. $s_j \trianglerighteq_{emb} t$

Example

$f(f(h(h(a)), h(x)), f(h(x), a)) \trianglerighteq_{emb} f(f(a, x), x)$

Remark

- Intuitively, we can establish an injection between the positions/symbols of the term on the left and those of the right. For example, we can map the blue f on the left into the blue f on the right, etc, and we erase/discard all the black symbols.
- Another way to understand $\trianglerighteq_{emb}$ is by the equality $\to^*_{\mathcal{R}_{emb}} = \trianglerighteq_{emb}$, where the system $\mathcal{R}_{emb} = \{f(x_1, \ldots, x_n) \mapsto x_i \mid f/n \in \Sigma\}$ discards symbols by projecting subterms. Indeed, taking again the previous example we have $f(f(h(h(a)), h(x)), f(h(x), a)) \to_{\mathcal{R}_{emb}} f(f(h(h(a)), h(x)), f(x, a)) \to_{\mathcal{R}_{emb}}$ $f(f(h(h(a)), h(x)), x) \to_{\mathcal{R}_{emb}} f(f(h(a), h(x)), x) \to_{\mathcal{R}_{emb}} f(f(a, h(x)), x) \to_{\mathcal{R}_{emb}}$ $f(f(a, x), x)$.

Lemma (A)

The relation $\triangleright_{emb}$ is contained in every simplification order.

Lemma

If $>$ is a simplification order, then $>$ is a reduction order (and thus WF).

Proof.

Uses the famous Kruskal's Theorem. □

And the inverse?

Question

If $\succ$ is a reduction order, then $\succ$ is a simplification order.

Let $\mathcal{R} = \{f(f(x)) \mapsto f(g(f(x)))\}$.
The system $\mathcal{R}$ terminates (exercise).
Thus $\rightarrow_{\mathcal{R}}^{+}$ is a reduction order by Theorem (Termination and Reduction Order).
Suppose that $\rightarrow_{\mathcal{R}}^{+}$ is also a simplification order.
Then by Lemma (A) $f(g(f(x))) \triangleright_{emb} f(f(x))$ implies
$f(g(f(x))) \rightarrow_{\mathcal{R}}^{+} f(f(x)) \rightarrow_{\mathcal{R}}^{+} f(g(f(x))) \ldots$.
Contradicts the fact that $\mathcal{R}$ is terminating.

Let $\gtrsim_\Sigma$ be a pre-order (reflexive and transitive) over a signature Σ. We associate to each symbol $f \in \Sigma$ a status in the set $\{\mathtt{LEX}, \mathtt{MUL}\}$ such that if $f \sim_\Sigma g$, then

- f and g have the same status,
- and if this status is LEX, then f and g have the same arity.

We note $f \in \Sigma_{\mathtt{LEX}}$ (resp. $f \in \Sigma_{\mathtt{MUL}}$) to indicate that $f \in \Sigma$ has LEX (resp. MUL) status. Thus $\Sigma = \Sigma_{\mathtt{LEX}} \uplus \Sigma_{\mathtt{MUL}}$.

The order $\succ_{\mathtt{rpo}}$

Let $\gtrsim_\Sigma$ be a pre-order over a signature Σ such that $\succ_\Sigma$ is WF. The RPO is defined as $s \succ_{\mathtt{rpo}} t$ iff one of the following cases hold

1. $s = f(s_1, \dots, s_n)$ and $\exists i$ s.t. $s_i \succ_{\mathtt{rpo}} t$ or $s_i = t$ or
2. $s = f(s_1, \dots, s_n)$, $t = g(t_1, \dots, t_m)$ and one of the following conditions is verified
   (a) $f \succ_\Sigma g$ and for all j, $s \succ_{\mathtt{rpo}} t_j$
   (b) $f \sim_\Sigma g$ have `MUL` status and $\{s_1, \dots, s_n\}(\succ_{\mathtt{rpo}})_{mul}\{t_1, \dots, t_m\}$.
   (c) $f \sim_\Sigma g$ have `LEX` status and $n = m$ and $(s_1, \dots, s_n)(\succ_{\mathtt{rpo}})_{lex}(t_1, \dots, t_n)$ and for all j, $s \succ_{\mathtt{rpo}} t_j$

Alternative definition of RPO

$$\frac{\exists i\ (s_i \succ_{\mathtt{rpo}} t \text{ or } s_i = t)}{f(s_1,\ldots,s_n) \succ_{\mathtt{rpo}} t}\ [1]$$

$$\frac{f \succ_\Sigma g \text{ and } \forall j\ s \succ_{\mathtt{rpo}} t_j}{s = f(s_1,\ldots,s_n) \succ_{\mathtt{rpo}} g(t_1,\ldots,t_m)}\ [2.a]$$

$$\frac{f \sim_\Sigma g \in \Sigma_{\mathtt{MUL}} \text{ and } \{s_1,\ldots,s_n\}\ (\succ_{\mathtt{rpo}})_{mul}\ \{t_1,\ldots,t_m\}}{s = f(s_1,\ldots,s_n) \succ_{\mathtt{rpo}} g(t_1,\ldots,t_m) = t}\ [2.b]$$

$$\frac{f \sim_\Sigma g \in \Sigma_{\mathtt{LEX}} \text{ and } (s_1,\ldots,s_n)(\succ_{\mathtt{rpo}})_{lex}\ (t_1,\ldots,t_n) \text{ and } \forall j\ s \succ_{\mathtt{rpo}} t_j}{s = f(s_1,\ldots,s_n)\ \succ_{\mathtt{rpo}} g(t_1,\ldots,t_n) = t}\ [2.c]$$

Remarks

- Is this definition well-founded?
  If $\succ_{\mathtt{rpo}}$ is defined on two terms of size k and l resp., then the recursive calls of
  - (1) use $\succ_{\mathtt{rpo}}$ on (k', l) with $k' < k$.
  - (2.a) use $\succ_{\mathtt{rpo}}$ on (k, l') with $l' < l$.
  - (2.b) use $(\succ_{\mathtt{rpo}})_{mul}$ on $\{k_1, \ldots, k_n\}$ and $\{l_1, \ldots, l_m\}$ with $\sum_i k_i < k$ and $\sum_j l_j < l$.
  - (2.c) use $(\succ_{\mathtt{rpo}})_{lex}$ on $(k_1, \ldots, k_n)$ and $(l_1, \ldots, l_m)$ with $\sum_i k_i < k$ and $\sum_j l_j < l$, and also $\succ_{\mathtt{rpo}}$ on (k, l_j) with $l_j < l$.
- Can we avoid condition $s \succ_{\mathtt{rpo}} t_j$ in case [2.c]?
  We would have that $a >_\Sigma a'$ implies $f(a, b) \succ_{\mathtt{rpo}} f(a', f(a, b)) \succ_{\mathtt{rpo}} f(a, b)$
- If all the symbols are `LEX`, the order is known as *LPO*.
- If all the symbols are `MUL`, the order is known as *MPO*.

Property of $\succ_{\mathtt{rpo}}$

Theorem

If $\succ_\Sigma$ is WF, then the relation $\succ_{\mathtt{rpo}}$ is a WF order.

Theorem

If $\succ_\Sigma$ is WF, then the associated relation $\succ_{\mathtt{rpo}}$ is a reduction order.

- As a consequence, to prove that a given rewriting system $\mathcal{R}$ is SN, it is sufficient to find an order $\succ_{\mathtt{rpo}}$ such that $l \succ_{\mathtt{rpo}} r$ for every $l \mapsto r \in \mathcal{R}$.
- The RPO was extended to the higher-order case.

Simple example

$$\Sigma = \{0, s, +, *\} \qquad \mathcal{R} = \begin{cases} 0 + y & \mapsto_{r1} & y \\ s(x) + y & \mapsto_{r2} & s(x + y) \\ 0 * y & \mapsto_{r3} & 0 \\ s(x) * y & \mapsto_{r4} & (x * y) + y \end{cases}$$

- Define $+ \succ_\Sigma s$, $* \succ_\Sigma +$ and $* \succ_\Sigma 0$, all with MUL (or LEX) status.
- Show that $l >_{\mathtt{rpo}} r$ for each rule $l \mapsto r \in \mathcal{R}$.

Thus for example for rule $s(x) * y \mapsto_{r4} (x * y) + y$

$$
\dfrac{
* \succ_{\Sigma} + \qquad
\dfrac{
* \sim_{\Sigma} * \in \Sigma_{\mathtt{MUL}} \qquad
\dfrac{
\dfrac{x = x}{s(x) \succ_{\mathtt{rpo}} x}
}{
\{s(x), y\} \; (\succ_{\mathtt{rpo}})_{mul} \; \{x, y\}
}
}{
s(x) * y \succ_{\mathtt{rpo}} x * y
}
\qquad
\dfrac{y = y}{s(x) * y \succ_{\mathtt{rpo}} y}
}{
s(x) * y \succ_{\mathtt{rpo}} (x * y) + y
}
$$

More subtle example

$$\Sigma = \{a/0, f/2, g/2\} \qquad \mathcal{R} = \begin{cases} f(g(x,y),z) & \mapsto_{r1} & f(x,y) \\ f(g(a,a),y) & \mapsto_{r2} & f(a, g(a,a)) \end{cases}$$

- Define a pre-order on $\Sigma = \{f, g, a\}$, and give the MUL status to all the symbols.
- Try to show that $l >_{\mathtt{rpo}} r$ for each rule $l \mapsto r \in \mathcal{R}$.
- Change the symbol f to LEX status.
- Start again to show $l >_{\mathtt{rpo}} r$ for each rule $l \mapsto r \in \mathcal{R}$.

Famous example: cut elimination in intuitionistic logic

$$
\begin{array}{lcl}
x[x/t] & \mapsto & t \\
y[x/t] & \mapsto & y \\
(\lambda z.u)[x/t] & \mapsto & \lambda z.u[x/t] \\
(y \texttt{ of } u \texttt{ is } w \texttt{ in } v)[x/t] & \mapsto & y \texttt{ of } u[x/t] \texttt{ is } w \texttt{ in } v[x/t] \\
({\color{red}x} \texttt{ of } u \texttt{ is } w \texttt{ in } v)[{\color{red}x}/y] & \mapsto & y \texttt{ of } u[x/y] \texttt{ is } w \texttt{ in } v[x/y] \\
({\color{red}x} \texttt{ of } u \texttt{ is } w \texttt{ in } v)[{\color{red}x}/\lambda z.t] & \mapsto & v[x/\lambda z.t][w/t[z/u[x/\lambda z.t]]] \\
({\color{red}x} \texttt{ of } u \texttt{ is } w \texttt{ in } v)[{\color{red}x}/x' \texttt{ of } t' \texttt{ is } z \texttt{ in } t] & \mapsto & x' \texttt{ of } t' \texttt{ is } z \texttt{ in } ((x \texttt{ of } u \texttt{ is } w \texttt{ in } v)[x/t])
\end{array}
$$

Combining orders

Suppose two SN relations $\mathcal{R}_1$ and $\mathcal{R}_2$. What about $\mathcal{R}_1 \cup \mathcal{R}_2$?

Famous counter-example by Toyama:

$$\mathcal{R}_1 = \{f(x,a,b) \mapsto f(x,x,x)\}$$

$$\mathcal{R}_2 = \left\{ \begin{array}{l} g(x,y) \mapsto x \\ g(x,y) \mapsto y \end{array} \right\}$$

The systems $\mathcal{R}_1$ and $\mathcal{R}_2$ (which do not share symbols!) are SN but $\mathcal{R}_1 \cup \mathcal{R}_2$ is not:

$$\begin{array}{l} f(g(a,b),g(a,b),g(a,b)) \rightarrow_{\mathcal{R}_2} f(g(a,b),a,g(a,b)) \rightarrow_{\mathcal{R}_2} \\ f(g(a,b),a,b) \rightarrow_{\mathcal{R}_1} f(g(a,b),g(a,b),g(a,b)) \rightarrow \ldots \end{array}$$

Termination by Postponement

A relation $\mathcal{R}$ can be postponed w.r.t. a relation $\mathcal{S}$ iff

$$\begin{array}{llllll} \text{for all } s,t,u \text{ s.t.} & s & \to_{\mathcal{R}} & t & \to_{\mathcal{S}} & u \\ \text{there is } v & s & \to_{\mathcal{S}}^{+} & v & \to_{\mathcal{R}\cup\mathcal{S}}^{*} & u \end{array}$$

Theorem

Let $\mathcal{R}$ and $\mathcal{S}$ be two WF relations s.t. $\mathcal{R}$ can be postponed w.r.t. $\mathcal{S}$. Then the relation $\mathcal{R} \cup \mathcal{S}$ is WF.

Corollary : If $\mathcal{S}$ is WF, then $\mathcal{S} \cup \triangleright$ is WF.

Proof of Termination by Postponement

Proof.

We will show that any $(\mathcal{R} \cup \mathcal{S})$-reduction sequence starting with an arbitrary term s is finite. We reason by (lexicographic) induction on (s, n), where s is compared using the WF relation $\mathcal{S}$, and n is the number of $\mathcal{R}$-steps separating s from the first $\mathcal{S}$-step.

- The base case is $(s, 0)$, where s is an $\mathcal{S}$-normal form. The sequence is then empty, so finite.
- If the $(\mathcal{R} \cup \mathcal{S})$-sequence does not contain any $\mathcal{S}$-step, then it is finite by WF of $\mathcal{R}$.
- If the $(\mathcal{R} \cup \mathcal{S})$-sequence does not contain any $\mathcal{R}$-step, then it is finite by WF of $\mathcal{S}$.
- If the $(\mathcal{R} \cup \mathcal{S})$-sequence starts with $n = k + 1$ $\mathcal{R}$-steps, it is of the form $s \rightarrow_{\mathcal{R}}^{k} s' \rightarrow_{\mathcal{R}} t \rightarrow_{\mathcal{S}} u \ldots$. The postponement hypothesis gives a sequence $s \rightarrow_{\mathcal{R}}^{k} s' \rightarrow_{\mathcal{S}}^{+} v \rightarrow_{\mathcal{R} \cup \mathcal{S}}^{*} u \ldots$. Since $(s, k) <_{lex} (s, k + 1)$, then this sequence is finite by the *i.h.*
- If the $(\mathcal{R} \cup \mathcal{S})$-sequence starts with $s \rightarrow_{\mathcal{S}} t \ldots$, then the sequence starting at t is smaller for the given order than the original one, *i.e.* $(t, m) <_{lex} (s, n)$ for any m. Then, the *i.h.* applies to the sequence starting at t, and thus the sequence starting at t, and thus the one starting at s is finite.

□

Example

Consider the simply typed λ-calculus enriched with a constant $\top$ type, equipped with the following rules:

$$
\begin{array}{llll}
(\lambda x.t)u & \mapsto_\beta & t\{x\backslash u\} & \\
\lambda x.t\,x & \mapsto_\eta & t & \text{if } x \notin \mathtt{fv}(t) \\
t & \mapsto_\Omega & \star & \text{if } \begin{cases} t \text{ is of } \top \text{ type} \\ t \neq \star \end{cases}
\end{array}
$$

Let $\mathcal{R} = \eta \cup \Omega$ and $\mathcal{S} = \beta$. Now,

- Show that $\eta \cup \Omega$ is WF.
- Show that $\eta \cup \Omega$ can be postponed w.r.t. β.
- Since β is SN, then conclude that $\eta \cup \Omega \cup \beta$ is SN.

Termination by projection/simulation

Theorem

Let $\mathcal{R}_1, \mathcal{R}_2$ be two relations over O s.t.

1. *$\mathcal{R}_2$ terminates*
2. *There is a simulation $\mathcal{T} : O \to O'$ and a relation $\mathcal{S}$ over O' s.t.*
   (a) $a \to_{\mathcal{R}_1} b$ *implies* $\mathcal{T}(a) \to^+_{\mathcal{S}} \mathcal{T}(b)$,
   (b) $a \to_{\mathcal{R}_2} b$ *implies* $\mathcal{T}(a) \to^*_{\mathcal{S}} \mathcal{T}(b)$.
   (c) $\mathcal{S}$ *terminates.*

Then, $(\mathcal{R}_1 \cup \mathcal{R}_2)$ also terminates.

Remark In particular, one can take $\mathcal{T}(a) = \mathcal{T}(b)$ in item 2(b).

Proof.

Suppose $(\mathcal{R}_1 \cup \mathcal{R}_2)$ does not termine. Since $\mathcal{R}_2$ terminates by hypothesis (1), every infinite $(\mathcal{R}_1 \cup \mathcal{R}_2)$-sequence necessarily contains an infinite $\mathcal{R}_1$-sequence. We can then write this sequence as:

$$a_1 \to^*_{\mathcal{R}_2} a_2 \to^+_{\mathcal{R}_1} a_3 \to^*_{\mathcal{R}_2} a_4 \to^+_{\mathcal{R}_1} \ldots$$

By the simulation hypothesis 2(a) and 2(b) we obtain:

$$\mathcal{T}(a_1) \to^*_{\mathcal{S}} \mathcal{T}(a_2) \to^+_{\mathcal{S}} \mathcal{T}(a_3) \to^*_{\mathcal{S}} \mathcal{T}(a_4) \to^+_{\mathcal{S}} \ldots$$

which contradicts termination of $\mathcal{S}$ which is hypothesis 2(c). □

(Famous) Example

Consider simply typed extensional λ-calculus

$$
\begin{array}{llll}
(\lambda x.M)\ N & \mapsto_{\beta} & M\{x \backslash N\} & \\
\pi_1\langle M, N\rangle & \mapsto_{\pi_1} & M & \\
\pi_2\langle M, N\rangle & \mapsto_{\pi_2} & N & \\
M & \mapsto_{\eta_{exp}} & \lambda x.Mx & \text{if} \begin{cases} M \text{ is of functional type} \\ M \text{ is not a } \lambda\text{-abstraction} \\ M \text{ is not applied in } C[M] \end{cases} \\
M & \mapsto_{sp_{exp}} & \langle \pi_1(M), \pi_2(M)\rangle & \text{if} \begin{cases} M \text{ is of product type} \\ M \text{ is not a pair} \\ M \text{ is not projected in } C[M] \end{cases}
\end{array}
$$

Thus for example if $z : A \times B$ and $x : (A \times B) \to (C \to D)$, then

$$I\ x\ z \to_{\beta}\ \ x\ z\ \ \to_{sp_{exp}}\ \ x\ \langle \pi_1(z), \pi_2(z) \rangle \to_{\eta_{exp}}\ \ \lambda y.(x\ \langle \pi_1(z), \pi_2(z) \rangle)\ y$$

Let $\mathcal{R}_1 = \beta \cup \pi_1 \cup \pi_2$ and $\mathcal{R}_2 = \eta_{exp} \cup sp_{exp}$ and $\mathcal{S} = \beta \cup \pi_1 \cup \pi_2$. Now,

- Show that $\eta_{exp} \cup sp_{exp}$ is terminating.
- Show that $\beta \cup \pi_1 \cup \pi_2$ is terminating (done).
- Show that $\eta_{exp} \cup sp_{exp}$ is also confluent.
- Define $\mathcal{T}(t)$ as the $\eta_{exp} \cup sp_{exp}$-normal form of t.
- Show that $t \to_{\beta \cup \pi_1 \cup \pi_2} t'$ implies $\mathcal{T}(t) \to^{+}_{\beta \cup \pi_1 \cup \pi_2} \mathcal{T}(t')$
- Show that $t \to_{\eta_{exp} \cup sp_{exp}} t'$ implies $\mathcal{T}(t) = \mathcal{T}(t')$ (evident).
- Conclude that all the system $\mathcal{R}_1 \cup \mathcal{R}_2$ is SN.

Termination by Dependency Pairs

- The technique is due to Aarts and Giesl.
- The order does not decrease for every step, but for the dependent ones.
- The technique is very suitable for functional programming.
- It was extended to higher-order by Sakai and Kusakari.
- It was extended to abstract rewriting by Lengrand.