Termination
Motivations

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:

$$
\text{end}(i, n) \equiv 1 \text{ if } a_i(n) \text{ terminates} \\
\text{end}(i, n) \equiv 0 \text{ if } a_i(n) \not\text{ terminates}
$$

$$
\text{Diag}(i) \equiv \begin{cases} 
\text{loop} & \text{if } \text{end}(i, i) = 1 \\
\text{stop} & \text{else}
\end{cases}
$$

For every $i$, $\text{Diag}(i)$ terminates iff $a_i(i)$ does not terminate.

But $\text{Diag}$ is an algorithm, so that $\exists a_j$ s.t. $\text{Diag} = a_j$. We then have $\text{Diag}(j)$ 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 $\text{end}$. 
Termination of a very simple system

\[ R = \{ f(g(x), y) \mapsto f(y, y) \} \]

is not even trivial!

\[ f(g(a), g(a)) \rightarrow f(g(a), g(a)) \rightarrow f(g(a), g(a)) \rightarrow \ldots \]
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 $p \in \text{Pos}(u)$.

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 $\theta$.

A relation $R$ is **well-founded** (WF) is there is no **infinite** sequence of the form 
$a_1 R a_2 R a_3 R \ldots$
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 $>$ over a signature $\Sigma$ is a reduction order iff

1. $>$ is stable by context
2. $>$ is stable by substitution
3. $>$ 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 $>$ s.t. $l > r$ 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 $> := \rightarrow^+_\mathcal{R}$.

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

Does $\mathcal{R}$ terminate?

$$\mathcal{R} = \left\{ \begin{array}{c}
\text{por}(x, t) \mapsto t \\
\text{por}(t, x) \mapsto t 
\end{array} \right\}$$

The number of symbol decreases....

- Let $|v|$ be the size of the term $v$. Consider the following order "$s >_1 t$ iff $|s| > |t|$".
  - We have $\text{por}(x, t) >_1 t$ and $\text{por}(t, x) >_1 t$.
  - But $>_1$ is not a reduction order since $>_1$ is not stable by substitution:
    $$\text{por}(x, \text{por}(y, t)) >_1 \text{por}(y, y) \text{ but } \text{por}(t, \text{por}(\text{por}(t, t), t)) \not>_1 \text{por}(\text{por}(t, t), \text{por}(t, t)).$$

- Let $|v|_x$ be the number of free occurrences of the variable $x$ in $v$. Consider the following order "$s >_2 t$ iff $|s| > |t|$ and for every variable $x$ $|s|_x \geq |t|_x$".
  - We have $\text{por}(x, t) >_2 t$ and $\text{por}(t, x) >_2 t$.
  - And $>_2$ is a reduction order.

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.

Definition

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

$s > t$ iff $\sigma(s) >_{\mathcal{A}} \sigma(t)$ for all homomorphisms $\sigma : T(X, \Sigma) \rightarrow \mathcal{A}$

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

Theorem

Let $\mathcal{A}$ be a $\Sigma$-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 $>$ is a reduction order.
Example: polynomial orders

A polynomial is an expression consisting of variables (indeterminates), addition, subtraction, multiplication, and non-negative integer exponentiation of variables. Examples of a polynomial are $2 \cdot x - 4 \cdot x + 7$ and $3 \cdot x + 2 \cdot x \cdot y \cdot z - y \cdot z + 1$.

A polynomial $\Sigma$-algebra $P_{\mathbb{N}}$ is defined by:

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

Example

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

Problem

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

A polynomial \( P \) is said to be \textit{completely monotonic} iff \( P \) depends on all its indeterminates and does not contain subtraction.

Example

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

Theorem

Let \( \mathcal{P}_\mathbb{N} \) be a polynomial \( \Sigma \)-algebra. If every \( f^{\mathcal{P}_\mathbb{N}} \) is a \textit{completely monotonic} polynomial, then the order \( > \) associated to \( >_{\mathcal{P}_\mathbb{N}} \) is a reduction order.
How does it work?

Does \( \mathcal{R} \) terminate?

\[
\mathcal{R} = \left\{ f(x, g(y, z)) \rightarrow g(f(x, y), f(x, z)) \right\}
\]

1. Define a polynomial for every function symbol: \( P_f(x, y) = x.y \) et \( P_g(x, y) = 2.x + y + 1 \).

2. Prove that \( f(x, g(y, z)) > g(f(x, y), f(x, z)) \): this is equivalent to prove that \( \sigma(x).(2.\sigma(y) + \sigma(z) + 1) \geq_{\mathcal{P}_\mathbb{N}} 2.\sigma(x).\sigma(y) + \sigma(x).\sigma(z) + 1 \) holds for every \( \sigma \) mapping \( x, y, z \) to the domain to be defined.

3. Define the domain as the one in which all the inequalities are valid: \( \mathcal{D} = \mathbb{N} - \{0, 1\} \subseteq \mathbb{N}^+ \).
Lexicographic order - particular case

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

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

Example

\[(4, "abc") >_{\text{lex}} (3, "abc") >_{\text{lex}} (2, "abcde") >_{\text{lex}} (2, "bcde") >_{\text{lex}} (2, "e") >_{\text{lex}} (1, "e") >_{\text{lex}} (0, \epsilon)\]
Lexicographic order - General case

If every $>_{A_i}$ is a strict order over the set $A_i$, then $>_{lex}$ is a strict order over $A_1 \times \ldots \times A_n$ defined as follows:

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

**Theorem**

Every order $>_{A_i}$ over $A_i$ is well-founded iff the lexicographic order $>_{lex}$ over $A_1 \times \ldots \times A_n$ is well-founded.
How does it work?

Does the following program terminate?

\[
\begin{align*}
\text{ackerman}(0, n) & \mapsto n + 1 \\
\text{ackerman}(m+1, 0) & \mapsto \text{ackerman}(m, 1) \\
\text{ackerman}(m+1, n+1) & \mapsto \text{ackerman}(m, \text{ackerman}(m+1, n))
\end{align*}
\]

\textbf{Proof.}

We show that \text{ackerman}(m, n) terminates by induction on \((m, n)\) w.r.t. the lexicographic order.
Does the following program terminate?

$$\mathcal{R} = \left\{ \begin{array}{c} f(f(x)) \mapsto g(f(x)) \\ g(g(x)) \mapsto f(x) \end{array} \right\}$$

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 $\mathcal{M} : \mathcal{A} \rightarrow \mathbb{N}$. 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

$\{a, a, b\} \uplus \{a, a, b, b\} = \{a, a, a, a, b, b\}$.
Multi-set order

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

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

Example

$\{5, 3, 1, 1\} >_\text{mul} \{4, 3, 3, 1\}$. Since $\{5, 3, 1, 1\} >_\text{mul} \{4, 3, 3, 1, 1\} >_\text{mul} \{4, 3, 3, 1\}$

Exercise: If $>$ is a strict order, then $>_\text{mul}$ is a strict order.

Theorem

Let $>$ be a strict order over $\mathcal{A}$, then $>$ is WF iff $>_\text{mul}$ 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
A simplification order over $\mathcal{F}(X, \Sigma)$ is an order $\succ$ s.t.

1. $\succ$ is stable by context
2. $\succ$ is stable by substitution
3. $t \succ u$ implies $t \succ u$
Example: embedding

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

- $s = x \triangleright_{emb} x = t$
- $s = f(s_1, \ldots, s_n) \triangleright_{emb} f(t_1, \ldots, t_n) = t$
- $s = f(s_1, \ldots, s_n) \triangleright_{emb} t$

Example

$s = f(f(h(h(a)), h(x)), f(h(x), a)) \triangleright_{emb} f(f(a, x), x) = t$ since $s \neq t$ and

- $a \triangleright_{emb} a$
- $h(a) \triangleright_{emb} a$
- $h(h(a)) \triangleright_{emb} a$
- $f(h(h(a)), h(x)) \triangleright_{emb} f(a, x)$
- $f(h(h(a)), h(x)), f(h(x), a)) \triangleright_{emb} f(f(a, x), x)$
Remarks

- 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 \( \succeq_{\text{emb}} \) is by the equality \( \rightarrow^*_{\mathcal{R}_{\text{emb}}} = \succeq_{\text{emb}} \), where the system \( \mathcal{R}_{\text{emb}} = \{ f(x_1, \ldots, x_n) \mapsto x_i \mid f/n \in \Sigma \} \) discards symbols by projecting subterms. Indeed, taking again the previous example we have

\[
\begin{align*}
  s &= f(f(h(h(a)), h(x)), f(h(x), a)) \\
  f(f(h(h(a)), h(x)), f(x, a)) &\rightarrow_{\mathcal{R}_{\text{emb}}} \\
  f(f(h(h(a)), h(x)), x) &\rightarrow_{\mathcal{R}_{\text{emb}}} \\
  f(f(h(a), h(x)), x) &\rightarrow_{\mathcal{R}_{\text{emb}}} \\
  f(f(a, h(x)), x) &\rightarrow_{\mathcal{R}_{\text{emb}}} \\
  f(f(a, x), x) &= t
\end{align*}
\]

Thus, \( s \succ_{\text{emb}} t \).
Lemma (A)

The relation \( \succ_{emb} \) is contained in every simplification order \( \succ \), i.e. if \( s \succ_{emb} t \) and \( \succ \) is a simplification order, then \( s \succ t \).

Lemma

If \( \succ \) is a simplification order, then \( \succ \) is a reduction order (and thus WF).

Proof.

Uses the famous Kruskal’s Theorem.
And the inverse?

**Question**

Is every reduction order also 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) $f(g(f(x))) \succ_{emb} f(f(x))$ 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.

**Conclusion**: there are reduction orders that are not simplification orders.
An Example of Simplification Order: The Recursive Path Ordering

Let $\succsim_{\Sigma}$ be a pre-order (reflexive and transitive) over a signature $\Sigma$. We associate to each symbol $f \in \Sigma$ a status in the set $\{\text{LEX}, \text{MUL}\}$ s.t. if $f \sim g$, then

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

We note $f \in \Sigma_{\text{LEX}}$ (resp. $f \in \Sigma_{\text{MUL}}$) to indicate that $f \in \Sigma$ has LEX (resp. MUL) status. Thus $\Sigma = \Sigma_{\text{LEX}} \cup \Sigma_{\text{MUL}}$. 
The order $>_\text{rpo}$

Let $\succeq_\Sigma$ be a pre-order over a signature $\Sigma$ such that $>_\Sigma$ is WF. The relation $s >_\text{rpo} t$ holds iff one of the following cases holds:

1. **[sub-term]** $s = f(s_1, \ldots, s_n)$ and $\exists i$ s.t. $s_i >_\text{rpo} t$ or $s_i = t$ or
2. **[Two symbols]** $s = f(s_1, \ldots, s_n)$, $t = g(t_1, \ldots, t_m)$ and one of the following conditions is verified:
   - (a) **[precedence]** $f >_\Sigma g$ and for all $j$, $s >_\text{rpo} t_j$
   - (b) **[multi-set]** $f \sim_\Sigma g$ have MUL status and $\{s_1, \ldots, s_n\} >_\text{rpo} \text{mul}\{t_1, \ldots, t_m\}$.
   - (c) **[lexicographic]** $f \sim_\Sigma g$ have LEX status and $n = m$ and $(s_1, \ldots, s_n) >_\text{rpo} \text{lex}(t_1, \ldots, t_n)$ and for all $j$, $s >_\text{rpo} t_j$
Alternative definition of RPO

\[
\exists i. (s_i >_{rpo} t \text{ or } s_i = t) \quad [1]
\]

\[
f(s_1, \ldots, s_n) >_{rpo} t
\]

\[
f >_{\Sigma} g \text{ and } \forall j. s >_{rpo} t_j \quad [2.a]
\]

\[
s = f(s_1, \ldots, s_n) >_{rpo} g(t_1, \ldots, t_m)
\]

\[
f \sim_{\Sigma} g \in \Sigma_{MUL} \text{ and } \{s_1, \ldots, s_n\} >_{rpo\text{\_mul}} \{t_1, \ldots, t_m\} \quad [2.b]
\]

\[
s = f(s_1, \ldots, s_n) >_{rpo} g(t_1, \ldots, t_m) = t
\]

\[
f \sim_{\Sigma} g \in \Sigma_{LEX} \text{ and } (s_1, \ldots, s_n) >_{rpo\text{\_lex}} (t_1, \ldots, t_n) \text{ and } \forall j. s >_{rpo} t_j \quad [2.c]
\]

\[
s = f(s_1, \ldots, s_n) >_{rpo} g(t_1, \ldots, t_n) = t
\]
Remarks

- Is this definition well-founded?
  If \( >_{rpo} \) is defined on two terms of size \( k \) and \( l \) resp., then the recursive calls of

  1. [sub-term] use \( >_{rpo} \) on \((k', l)\) with \( k' < k \).
  2. [precedence] use \( >_{rpo} \) on \((k, l')\) with \( l' < l \).
  3. [multi-set] use \( >_{rpo,mul} \) on \(\{k_1, \ldots, k_n\}\) and \(\{l_1, \ldots, l_m\}\) with \( k_i < k \) and \( l_j < l \).
  4. [lexicographic] use \( >_{rpo,lex} \) on \((k_1, \ldots, k_n)\) and \((l_1, \ldots, l_m)\) with \( k_i < k \) and \( l_j < l \), and also \( >_{rpo} \) on \((k, l_j)\) with \( l_j < l \).

- Can we avoid condition \( s >_{rpo} t_j \) in case [lexicographic]?
  We would have that \( a >_{\Sigma} a' \) implies \( f(a, b) >_{rpo} f(a', f(a, b)) >_{rpo} f(a, b) \).

- If all the symbols are LEX, the order is known as \( LPO \).
- If all the symbols are MUL, the order is known as \( MPO \).
Theorem

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

Theorem

If $\succ_{\Sigma}$ is WF, then the associated relation $\succ_{\text{rpo}}$ 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 $\succ_{\text{rpo}}$ such that $l \succ_{\text{rpo}} r$ for every $l \rightarrow r \in \mathcal{R}$.
- The RPO was extended to the higher-order case.
Simple example

\[ R = \begin{cases} 
0 + y & \mapsto_{r_1} y \\
 s(x) + y & \mapsto_{r_2} s(x + y) \\
 0 \ast y & \mapsto_{r_3} 0 \\
 s(x) \ast y & \mapsto_{r_4} (x \ast y) + y 
\end{cases} \]

- Define $+ \succ_{\Sigma} s$, $\ast \succ_{\Sigma} +$ and $\ast \succ_{\Sigma} 0$, all with MUL (or LEX) status.
- Show that $l \succ_{rpo} r$ for each rule $l \mapsto r \in R$. 
Thus for example for rule $s(x) \ast y \mapsto_{r4} (x \ast y) + y$

\[
\begin{array}{ccc}
\text{x} = \text{x} & \Rightarrow \text{s(x) >}_{\text{rpo}} \text{x} & \text{y} = \text{y} \\
\text{s(x) >}_{\text{rpo}} \text{x} & \Rightarrow \text{s(x) \ast y >}_{\text{rpo}} \text{x} \ast y & \text{s(x) \ast y >}_{\text{rpo}} \text{y} \\
\text{s(x) \ast y >}_{\text{rpo}} \text{x} \ast y & \Rightarrow \text{s(x) \ast y >}_{\text{rpo}} (x \ast y) + y
\end{array}
\]
More subtle example

\[ \mathcal{R} = \begin{cases} 
  f(g(x, y), z) & \mapsto_{r_1} f(x, y) \\
  f(g(a, a), y) & \mapsto_{r_2} f(a, g(a, a)) 
\end{cases} \]

- Define a pre-order on \( \{f, g, a\} \), and give the MUL status to all the symbols.
- Try to show that \( l >_{\text{rpo}} r \) for each rule \( l \mapsto r \in \mathcal{R} \).
- Change the symbol \( f \) to LEX status.
- Start again to show \( l >_{\text{rpo}} r \) for each rule \( l \mapsto r \in \mathcal{R} \).
Famous example: cut elimination in intuitionistic logic

\[
x[x/t] \quad \mapsto \quad t \\
y[x/t] \quad \mapsto \quad y \\
(\lambda z.u)[x/t] \quad \mapsto \quad \lambda z.u[x/t] \\
(y \text{ of } u \text{ is } w \text{ in } v)[x/t] \quad \mapsto \quad y \text{ of } u[x/t] \text{ is } w \text{ in } v[x/t] \\
(x \text{ of } u \text{ is } w \text{ in } v)[x/y] \quad \mapsto \quad y \text{ of } u[x/y] \text{ is } w \text{ in } v[x/y] \\
(x \text{ of } u \text{ is } w \text{ in } v)[x/\lambda z.t] \quad \mapsto \quad v[x/\lambda z.t][w/t[z/u[x/\lambda z.t]]] \\
(x \text{ of } u \text{ is } w \text{ in } v)[x/x' \text{ of } t' \text{ is } z \text{ in } t] \quad \mapsto \quad x' \text{ of } t' \text{ is } z \text{ in } ((x \text{ of } u \text{ is } w \text{ in } v)[x/t])
\]
Suppose two SN relations $R_1$ and $R_2$. What about $R_1 \cup R_2$?

Famous counter-example by Toyama:

$$R_1 = \{ f(x, a, b) \mapsto f(x, x, x) \}$$

$$R_2 = \begin{cases} 
  g(x, y) \mapsto x \\
  g(x, y) \mapsto y 
\end{cases}$$

The systems $R_1$ and $R_2$ (which do not share symbols!) are SN but $R_1 \cup R_2$ is not:

$$f(g(a, b), g(a, b), g(a, b)) \rightarrow_{R_2} f(g(a, b), a, g(a, b)) \rightarrow_{R_2}$$

$$f(g(a, b), a, b) \rightarrow_{R_1} f(g(a, b), g(a, b), g(a, b)) \rightarrow \ldots$$
A relation $R$ can be postponed w.r.t. a relation $S$ iff

for all $s, t, u$ s.t. $s \rightarrow_R t \rightarrow_S u$

there is $v$ $s \rightarrow^+_S v \rightarrow^*_{R \cup S} u$

**Theorem**

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

**Corollary** : If $S$ is WF, then $S \cup \triangleright$ is WF.
Proof.

We will show that any \((R \cup S)\)-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 \(S\), and \(n\) is the number of \(R\)-steps separating \(s\) from the first \(S\)-step.

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

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

\[
\begin{align*}
(\lambda x.t)u & \mapsto_\beta t\{x\backslash u\} \\
\lambda x.t & \mapsto_\eta t \quad \text{if } x \notin \text{fv}(t) \\
t & \mapsto_\Omega \star \quad \text{if } \begin{cases} 
&t \text{ is of } \top \text{ type} \\
&t \neq \star
\end{cases}
\end{align*}
\]

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

- Show that $\eta \cup \Omega$ is WF.
- Show that $\eta \cup \Omega$ can be postponed w.r.t. $\beta$.
- Since $\beta$ 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) $a \rightarrow_{\mathcal{R}_1} b$ implies $\mathcal{T}(a) \rightarrow^{+}_{\mathcal{S}} \mathcal{T}(b)$,
   
   (b) $a \rightarrow_{\mathcal{R}_2} b$ implies $\mathcal{T}(a) \rightarrow^{*}_{\mathcal{S}} \mathcal{T}(b)$.

(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:

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

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

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

which contradicts termination of $\mathcal{S}$ which is hypothesis 2(c). □
Consider simply typed extensional $\lambda$-calculus

$$(\lambda x.M) N \mapsto_{\beta} M \{x\} \{N\}$$

$$\pi_1\langle M, N \rangle \mapsto_{\pi_1} M$$

$$\pi_2\langle M, N \rangle \mapsto_{\pi_2} N$$

$$M \mapsto_{\eta_{exp}} \lambda x. Mx \quad \text{if } \begin{cases} M \text{ is of functional type} \\ M \text{ is not a } \lambda\text{-abstraction} \\ M \text{ is not applied in } C[M] \end{cases}$$

$$M \mapsto_{sp_{exp}} \langle \pi_1(M), \pi_2(M) \rangle \quad \text{if } \begin{cases} M \text{ is of product type} \\ M \text{ is not a pair} \\ M \text{ is not projected in } C[M] \end{cases}$$
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.