

---

---

## Termination

---

---

1

## Motivations

---

Termination is essential to proof  
correctness of programs.

But

Termination is an undecidable property.

2

## Undecidability of termination

---

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

$\text{end}(i, n)$     1 if  $a_i(n)$  terminates     $\text{Di ag}(i)$     if  $\text{end}(i, i) = 1$  then loop  
 $\text{end}(i, n)$     0 if  $a_i(n) \rightarrow$  terminates    else stop

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

But  $\text{Di ag}$  is an algorithm, so that  $a_j$  s.t.  $\text{Di ag} = a_j$ . We then have

$\text{Di ag}(j)$  terminates iff  $a_j(j)$  terminates, that is

$a_j(j)$  terminates iff  $a_j(j)$  does not terminate.

3

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

4

## Termination of a very simple system

---

$$f(g(x), y) \rightarrow f(y, y)$$

is not even trivial!

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

5

## Recall

---

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

A relation  $R$  over  $T(X, \Sigma)$  is **stable by substitution** iff  $t R t'$  implies  $(t) R (t')$  for every substitution  $\sigma$ .

7

## 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

6

## Termination by reduction orders

---

**Pre-order** : reflexive and transitive relation.

**Partial order** : reflexive, antisymmetric and transitive relation.

**Strict order** : irreflexive and transitive relation.

A strict order  $R$  over a signature  $\Sigma$  is a **reduction order** iff

1. Each symbol  $f \in \Sigma$  is monotone w.r.t  $R$
2.  $R$  is stable by substitution
3.  $R$  is WF

Why reduction orders are important?

8

**Theorem :** A rewriting system  $R$  terminates iff there exists a reduction order s.t.  $l \rightarrow r$  for every rewriting rule  $l \rightarrow r \in R$ .

9

### Interpretation as particular case of reduction order

The reduction order is first defined on the **interpretation** of terms, and not directly on terms.

Let  $\prec_A$  be a WF strict order over the domain of a  $\Sigma$ -algebra  $A$ .

**Définition :** The associated order over the terms is given by :

$s \prec t$  iff  $\Phi(s) \prec_A \Phi(t)$  for all homomorphisms  $\Phi : T(X, \Sigma) \rightarrow A$

**Theorem :** If for every  $f \in \Sigma$ , the interpretation  $f^A$  is monotone w.r.t.  $\prec_A$ , then  $\prec$  is a reduction order.

11

### How does it work ?

Does  $R$  terminate ?

$$R = \begin{array}{l} \text{por}(x, t) \rightarrow t \\ \text{por}(t, x) \rightarrow t \end{array}$$

The number of symbol decreases....

" $s \prec_1 t$  iff  $|s| > |t|$ " is not a reduction order (not stable by substitution) :

$\text{por}(x, \text{por}(y, t)) \prec_1 \text{por}(y, y)$  but  
 $\text{por}(t, \text{por}(\text{por}(t, t), t)) \prec_1 \text{por}(\text{por}(t, t), \text{por}(t, t))$ .

" $s \prec_2 t$  iff  $|s| > |t|$  and  $x \prec_{s/x} t/x$ " is a reduction order.

10

### Example : polynomial orders

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

- A domain which is a subset of  $\mathbb{N}^+$
- A polynomial  $P_f$  for each  $f/n \in \Sigma$ , there is s.t.  
 $f^{P_{\mathbb{N}}}(a_1, \dots, a_n) = P_f(a_1, \dots, a_n)$ .

**Example :** Let  $\Sigma = \{f/2, g/2, a/0\}$ . Consider the morphism  $\Phi$  given polynomials  $P_f(x, y) = x.y$  and  $P_g(x, y) = 2.x + y + 1$  and  $P_a = 2$ . Then we have  $\Phi(f(a, g(a, a))) = 2.(2.2 + 2 + 1)$ .

**Problem :** Polynomials are not necessarily monotone, for example if  $P_f(X, Y) = X^2$  we have  $3 > 2$  but  $P_f(2, 3) = 4 > 4 = P_f(2, 2)$ .

12

## Towards a polynomial order as interpretation

A polynomial  $P$  is **completely monotone** iff it depends on all its indeterminates.

**Example :**  $P(x, y) = 3.x + y + 2$  and  $P(x, y) = x.y$  are all completely monotone.

**Theorem :** Let  $P_{\mathbb{N}}$  be a polynomial  $\Sigma$ -algebra. If every  $f^{P_{\mathbb{N}}}$  is a completely monotone polynomial, then the order associated to  $P_{\mathbb{N}}$  is a reduction order.

13

## 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 :**

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

15

## How does it work ?

Does  $R$  terminate?

$$R = f(x, g(y, z)) \quad g(f(x, y), f(x, z))$$

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  $f(x, g(y, z)) \quad g(f(x, y), f(x, z))$  : Prove  
 $(x).(2.(y) + (z) + 1) \quad P_{\mathbb{N}} \quad 2.(x).(y) + (x).(z) + 1$   
for every  $(x), (y), (z)$  in some restriction of the domain.
3. Define the domain as the one in which all the inequalities are valid :  $\mathbb{N} - \{0, 1\}$ .

14

## Lexicographic order - General case

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

$$(x_1, \dots, x_n) >_{lex} (x'_1, \dots, x'_n) \text{ iff } \begin{matrix} 1 & j & n \\ (x_j >_{A_j} x'_j \text{ and } 1 & i < j & x_i = x'_i) \end{matrix}$$

**Theorem :** Every order  $>_{A_i}$  over  $A_i$  is well-founded iff the lexicographic order  $>_{lex}$  over  $A_1 \times \dots \times A_n$  is well-founded.

16

## How does it work ?

---

Does the following program terminate ?

$\text{ackerman}(0, n) \quad n+1$   
 $\text{ackerman}(m+1, 0) \quad \text{ackerman}(m, 1)$   
 $\text{ackerman}(m+1, n+1) \quad \text{ackerman}(m, \text{ackerman}(m+1, n))$

*Proof.* We show that  $\text{ackerman}(m, n)$  terminates by induction on  $(m, n)$  w.r.t. the lexicographic order. ■

17

## Multi-set order

---

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

**Example :**  $\{a, a, b\}$ .

Let  $M$  and  $N$  be two multi-sets. The **multi-set union** is defined by  $M \cup N(a) = M(a) + N(a)$ .

19

## Another example ?

---

Does the following program terminate ?

$f(f(x)) \quad g(f(x))$   
 $g(g(x)) \quad f(x)$

*Proof.* Show that  $t > u$  iff  $(/t/, /t|_f) >_{lex} (/u/, /u|_f)$  is a reduction order. ■

18

## Multi-set order

---

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

$M \text{ mul } \{x\} \text{ mul } M \text{ mul } \{y_1, \dots, y_n\}$ , where  $n \geq 0$  and  $x > y_i$ .

**Example :**  $\{5, 3, 1, 1\} \text{ mul } \{4, 3, 3, 1\}$ .

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

**Theorem :** Let  $>$  be a strict order over  $A$ , then  $\text{mul}$  is WF iff  $>$  is WF.

20

## 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.

- Represent the initial amount of money by a multi-set.
- Represent the daily activity of the man by a decreasing order on multi-sets.

21

## Simplification orders

---

A **simplification order** over  $T(X, \Sigma)$  is an order s.t.

1. All the symbols of  $\Sigma$  are monotone w.r.t
2. is stable by substitution
3.  $t \leq u$  implies  $t \leq v$

23

## Other known examples

---

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

22

## Example : embedding

---

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

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

**Example :**  $f(f(h(h(a)), h(x)), f(h(x), a)) \succeq_{emb} f(f(a, x), x)$

24

## Termination by simplification orders

---

**Lemma :** The relation  $\text{emb}$  is contained in every simplification order.

**Lemma :** If  $\succsim$  is a simplification order, then it is a reduction order (and thus WF).

*Proof.* Uses the famous Kruskal's Theorem.  $\blacksquare$

25

## Example : Recursive Path Ordering

---

Let  $\succsim_\Sigma$  be a pre-order over  $\Sigma$ . We associate to each symbol  $f \in \Sigma$  a **status** in  $\{LEX, MUL\}$  s.t. if  $f \succsim g$ , then

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

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

27

## And the inverse ?

---

Let  $R = f(f(x)) \rightarrow f(g(f(x)))$ .

The system  $R$  terminates (exercise).

Thus  $\stackrel{+}{R}$  is a reduction order.

Suppose that  $\stackrel{+}{R}$  is also a simplification order.

Then  $f(g(f(x))) \text{ emb } f(f(x))$  implies  
 $f(g(f(x))) \stackrel{+}{R} f(f(x)) \stackrel{+}{R} f(g(f(x))) \dots$

Contradiction with the termination of  $R$ .

26

## The order $\text{rpo}$

---

Let  $\succsim_\Sigma$  be a pre-order over a signature  $\Sigma$  such that  $\succsim_\Sigma$  is WF.

The **RPO** is given by  $s \text{ rpo } t$  iff

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

28

## Alternative definition of RPO

$$\frac{i (s_i \text{ rpo } t \text{ or } s_i = t)}{f(s_1, \dots, s_n) \text{ rpo } t} \quad [1]$$

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

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

$$\frac{f \text{ } \Sigma \text{ } g \text{ } \Sigma_{LEX} \text{ and } (s_1, \dots, s_n) ( \text{ rpo } )_{lex} (t_1, \dots, t_n) \text{ and } j \text{ } s \text{ rpo } t_j}{s = f(s_1, \dots, s_n) \text{ rpo } g(t_1, \dots, t_n) = t} \quad [2.]$$

29

## Property of $\text{rpo}$

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

**Theorem :** If  $\Sigma$  is WF, then the associated relation  $\text{rpo}$  is a reduction order.

As a consequence, to prove *SN* of a given system  $R$ , it is sufficient to find an order  $\text{rpo}$  such that  $l \text{ rpo } r$  for every  $l \text{ } r \text{ } R$ .

The RPO was extended to the higher-order case by Jouannaud and Rubio.

31

## Remarks

- Is this definition well-founded ?
- Can we avoid condition  $s \text{ rpo } t_j$  in case LEX [2.c] ?  
We would have that  $a \text{ } \Sigma \text{ } a$  implies  $f(a, b) \text{ rpo } f(a, f(a, b))$
- If all the symbols are LEX, the order is known as *LPO*.
- If all the symbols are MUL, the order is known as *MPO*.

30

## Simple example

$$R \quad \begin{array}{ll} 0 + y & r1 \text{ } y \\ s(x) + y & r2 \text{ } s(x + y) \\ 0 \text{ } y & r3 \text{ } 0 \\ s(x) \text{ } y & r4 \text{ } (x \text{ } y) + y \end{array}$$

- Define  $+$   $\Sigma$   $S$ ,  $\Sigma$   $+$  and  $\Sigma$   $0$ , all with MUL (or LEX) status.
- Show that  $l >_{\text{rpo}} r$  for each rule  $l \text{ } r \text{ } R$ .

32



Thus for example for rule  $s(x) \ y \ r_4 \ (x \ y) + y$

$$\frac{\Sigma + \frac{\Sigma \quad \frac{\frac{x = x}{s(x) \ rpo \ x} \quad \frac{\{s(x), y\} ( \ rpo)_{mul} \ \{x, y\}}{\Sigma}}{s(x) \ y \ rpo \ x \ y} \quad \frac{y = y}{s(x) \ y \ rpo \ y}}{s(x) \ y \ rpo \ (x \ y) + y}$$

33

### Famous example : cut elimination in intuitionistic logic

$x[x/t]$	$t$
$y[x/t]$	$y$
$(\lambda z.u)[x/t]$	$\lambda z.u[x/t]$
$(y \text{ of } u \text{ is } w \text{ in } v)[x/t]$	$y \text{ of } u[x/t] \text{ is } w \text{ in } v[x/t]$
$(x \text{ of } u \text{ is } w \text{ in } v)[x/y]$	$y \text{ of } u[x/y] \text{ is } w \text{ in } v[x/y]$
$(x \text{ of } u \text{ is } w \text{ in } v)[x/z.t]$	$v[x/\lambda z.t][w/t[z/u[x/\lambda z.t]]]$
$(x \text{ of } u \text{ is } w \text{ in } v)[x/x \text{ of } t \text{ is } z \text{ in } t]$	$x \text{ of } t \text{ is } z \text{ in } ((x \text{ of } u \text{ is } w \text{ in } v)[x/t])$

35

### More subtle example

$$R \quad \begin{array}{ll} f(g(x, y), z) & r_1 \ f(x, y) \\ f(g(a, a), y) & r_2 \ f(a, g(a, a)) \end{array}$$

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

34

### Combining orders

Suppose two SN relations  $R_1$  and  $R_2$ . What about  $R_1 \ R_2$ ?

Counter-example by Toyama :

$$\begin{array}{ll} R_1 = f(x, a, b) & f(x, x, x) \\ R_2 = g(x, y) & x \\ & g(x, y) \ y \end{array}$$

The systems  $R_1$  and  $R_2$  are SN but  $R_1 \ R_2$  is not :

$$\begin{array}{ll} f(g(a, b), g(a, b), g(a, b)) & R_2 \ f(g(a, b), a, g(a, b)) \ R_2 \\ f(g(a, b), a, b) & R_1 \ f(g(a, b), g(a, b), g(a, b)) \ \dots \end{array}$$

36

## Termination by postponement

A relation  $R$  can be **postponed** w.r.t. a relation  $S$  iff

for all  $s, t, u$  s.t.  $s \quad R \quad t \quad S \quad u$

there is  $v$   $s \quad S^+ \quad v \quad R \quad S \quad u$

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

**Corollary :** If  $S$  is WF, then  $S$  is WF.

37

## Termination by projection/simulation

**Theorem :** Let  $R_1, R_2$  be two relations over  $O$  s.t.

1.  $R_2$  terminates
2. There is a **simulation**  $T : O \rightarrow O$  and a **relation**  $S$  over  $O$  s.t.
  - $a \quad R_1 \quad b$  implies  $T(a) \quad S^+ \quad T(b)$ ,
  - $a \quad R_2 \quad b$  implies  $T(a) \quad S \quad T(b)$ .

Then, if  $S$  terminates,  $(R_1 \quad R_2)$  also terminates.

39

## Example

Consider **simply typed**  $\lambda$ -calculus with the following rules :

$$\begin{array}{lcl} ( \lambda x.M ) N & \beta & M\{x/N\} \\ x.M \quad x & \eta & M, \text{ if } x \not\in \text{fv}(M) \\ M & \Omega & , \text{ if } M = \end{array}$$

Let  $R = \Omega$  and  $S =$  . Now,

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

38

## (Famous) Example

Consider **simply typed** extensional  $\lambda$ -calculus

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

40

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

$$\lambda x z \beta \cdot x z \xrightarrow{sp_{exp}} x \pi_1(z), \pi_2(z) \xrightarrow{\eta_{exp}} y.(x \pi_1(z), \pi_2(z)) y$$

Let  $R_1 = \beta \rightarrow \beta$  and  $R_2 = \eta_{exp} \rightarrow sp_{exp}$  and

$S = \beta \rightarrow \beta$ . Now,

- Show that  $\eta_{exp} \rightarrow sp_{exp}$  is terminating.
- Show that  $\beta \rightarrow \beta$  is terminating (done).
- Show that  $\eta_{exp} \rightarrow sp_{exp}$  is also confluent.
- Define  $T(t)$  as the  $\eta_{exp} \rightarrow sp_{exp}$ -normal form of  $t$ .
- Show that  $t \beta \rightarrow \beta t$  implies  $T(t) \beta \rightarrow \beta T(t)$
- Show that  $t \eta_{exp} \rightarrow sp_{exp} t$  implies  $T(t) = T(t)$  (evident).
- Conclude that all the system  $R_1 \rightarrow R_2$  is SN.

41

## Termination by Dependency Pairs

---

- The technique is due to Aarts and Giesl.
- The order does not decrease for **every** step, but for the **dependent** one.
- 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.

42