Termination of Term Rewriting Systems

Termination

R (or $\rightarrow_R$) terminates
iff all derivation issued from any term terminates.

Termination implies the existence of normal form(s) for any term.

Termination is in general undecidable
but interesting sufficient conditions can be found.

Proving Termination

Termination of rewriting can be checked by sufficient conditions:

- Syntactic and semantic methods (applying directly to the TRS)
  KBO [Knuth & Bendix 1970], LPO [Kamin & Levy 1980], RPO [Dershowitz 1982], RPOS [Steinbach 1989], GPO [Dershowitz & Hoot 1995], Polynomial interpretations [Lankford 1975, Ben Cherifa & Lescanne 1986],...
- Transformational approaches (transforming one TRS into another)
  Semantic labelling [Zantema 1995], Dependency pairs [Arts & Giesl 1996], ...
- Induction on the derivation trees (schematization by abstraction and narrowing of the derivations)
  [Fissore & Gnaedig & Kirchner 2003]

Proving termination could be tricky . . .

Proving termination could be tricky ...

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

is terminating

Proving termination could be tricky ...

$$f(a, b, x) \quad \rightarrow \quad f(x, x, x)$$

is terminating

$$\begin{array}{rcl} g(x, y) & \rightarrow & x \\ g(x, y) & \rightarrow & y, \end{array}$$

is terminating

Proving termination could be tricky ...

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

is terminating

$$\begin{array}{rcl} g(x, y) & \to & x \\ g(x, y) & \to & y, \end{array}$$

is terminating

Is the union terminating?

Proving termination could be tricky ...

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

Proving termination could be tricky ...

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

We have the derivation:

$$
\begin{tikzcd}
f(g(a, b), g(a, b), g(a, b)) \arrow[r] & f(a, g(a, b), g(a, b)) \arrow[r] & f(a, b, g(a, b)) \arrow[ll, bend right]
\end{tikzcd}
$$

[Toyama 1986]

Abstract rewrite systems

- **Consider a set $\mathcal{T}$**
- **Consider a binary relation $\longrightarrow$ on $\mathcal{T}$ (one-step reduction)**
  - $a \longrightarrow b$: b is the reduct of a
- Induced relations
  - transitive closure: $\xrightarrow{+}$
  - transitive reflexive closure: $\xrightarrow{*}$
  - symetric closure: $\longleftrightarrow$

Normalization

Consider an ARS ($\mathcal{T}$,$\rightarrow$)

- An element $t \in \mathcal{T}$ is a $\rightarrow$-**normal form** if there exists no $t' \in \mathcal{T}$ such that $t \rightarrow t'$.

- The relation $\rightarrow$ is **terminating** (or **strongly normalizing**, or **noetherian**) if every reduction sequence is finite.

Normalization

Consider an ARS ($\mathcal{T}$,$\to$)

- An element $t \in \mathcal{T}$ is a $\to$-**normal form** if there exists no $t' \in \mathcal{T}$ such that $t \to t'$.

- The relation $\to$ is **terminating** (or **strongly normalizing**, or **noetherian**) if every reduction sequence is finite.

  $a \to a$ is not terminating

- The relation $\to$ is **weakly normalizing** (or weakly terminating) if every element $t \in \mathcal{T}$ has a normal form.

Normalization

Consider an ARS ($\mathcal{T}$,$\rightarrow$)

- An element $t \in \mathcal{T}$ is a $\rightarrow$-**normal form** if there exists no $t' \in \mathcal{T}$ such that $t \rightarrow t'$.

- The relation $\rightarrow$ is **terminating** (or **strongly normalizing**, or **noetherian**) if every reduction sequence is finite.

  $a \rightarrow a$ is not terminating

- The relation $\rightarrow$ is **weakly normalizing** (or weakly terminating) if every element $t \in \mathcal{T}$ has a normal form.

  $a \rightarrow a \quad a \rightarrow b$ is weakly terminating

- The relation $\rightarrow$ has the **unique normal form property** if for any $t, t' \in \mathcal{T}$, $t \overset{*}{\longleftrightarrow} t'$ and t, t' are normal forms imply $t = t'$.

Showing strong normalization

“Size”

A (partial) order on $\mathcal{T}$ is a reflexive, antisymetric and transitive relation.
An ordering is **total** on $\mathcal{T}$ when two terms are always comparable

An order $>$ is well-founded or Noetherian on $\mathcal{T}$ if there is no infinite decreasing sequence on $\mathcal{T}$:

$$t_1 > t_2 > t_3 > \ldots$$

“Size”

A (partial) order on $\mathcal{T}$ is a reflexive, antisymetric and transitive relation.
An ordering is **total** on $\mathcal{T}$ when two terms are always comparable

An order $>$ is well-founded or Noetherian on $\mathcal{T}$ if there is no infinite decreasing sequence on $\mathcal{T}$:

$$t_1 > t_2 > t_3 > \dots$$

Theorem

Consider an ARS $(\mathcal{A},\to)$.
$\to$ is terminating
iff
there exists a well-founded (partial) order $>$ on $\mathcal{T}$ and a mapping ϕ from $\mathcal{A}$ to $\mathcal{T}$ s.t.
for all rewrite rule $a \to a'$ implies $\phi(a) > \phi(a')$.

Example

Use the order $(>, \mathbb{N})$ which is well-founded.

Several choices for strings $\mathcal{A} = (\bullet \mid \circ)^*$

The rules of the game:

$$\begin{array}{rcl} \bullet\bullet & \rightarrow & \circ \\ \circ\circ & \rightarrow & \circ \\ \bullet\circ & \rightarrow & \bullet \\ \circ\bullet & \rightarrow & \bullet \end{array}$$

$l > r$ if $|l|_{\bullet\circ} > |r|_{\bullet\circ}$
($|t|_{\bullet\circ}$=number of $\bullet$ and $\circ$ of the term t built out of $\bullet$ and $\circ$)

Example

The rules of the game:

$$\begin{array}{ccc} \bullet\bullet & \to & \boxed{\circ\circ} \\ \circ\circ & \to & \circ \\ \bullet\circ & \to & \bullet \\ \circ\bullet & \to & \bullet \end{array}$$

Example

The rules of the game:

$$\begin{array}{ccc} \bullet\bullet & \rightarrow & \boxed{\circ\circ} \\ \circ\circ & \rightarrow & \circ \\ \bullet\circ & \rightarrow & \bullet \\ \circ\bullet & \rightarrow & \bullet \end{array}$$

$l > r$ if $|l|_{\bullet\circ+\bullet} > |r|_{\bullet\circ+\bullet}$

Example

The rules of the game:

$$\begin{aligned}
\bullet\bullet &\to \circ\circ \\
\circ\circ &\to \boxed{\bullet} \\
\bullet\circ &\to \bullet \\
\circ\bullet &\to \bullet
\end{aligned}$$

Example

The rules of the game:

$$\begin{array}{ccc} \bullet\bullet & \rightarrow & \circ\circ \\ \circ\circ & \rightarrow & \boxed{\bullet} \\ \bullet\circ & \rightarrow & \bullet \\ \circ\bullet & \rightarrow & \bullet \end{array}$$

$|I| = 2|I|_{\bullet\circ} + |I|_{\bullet}$

Reduction ordering

A *reduction ordering* is a well-founded ordering on $\mathcal{T}$, stable by context and substitution:

- for every context $C[\_]$ and for all substitutions σ,
  if $t > s$ then $C[t] > C[s]$ and $\sigma(t) > \sigma(s)$.

Theorem

R terminates iff there exists a reduction ordering $>$ s.t. for all rewrite rule $(l \to r) \in R$, $l > r$.

Reduction ordering

Is size a reduction ordering?

Reduction ordering

Is size a reduction ordering?

$|f(f(x,x),y)| > |f(y,y)|$
but
$|f(f(x,x),f(x,x))| \not> |f(f(x,x),f(x,x))|$

Reduction ordering

Is size a reduction ordering?

$|f(f(x,x),y)| > |f(y,y)|$
but
$|f(f(x,x),f(x,x))| \not> |f(f(x,x),f(x,x))|$

Prove that

$$t > s \text{ if } |t|_x > |s|_x \text{ for all variable } x$$

is a reduction ordering

Reduction ordering

Is size a reduction ordering?

$|f(f(x,x),y)| > |f(y,y)|$
but
$|f(f(x,x),f(x,x))| \not> |f(f(x,x),f(x,x))|$

Prove that

$$t > s \text{ if } |t|_x > |s|_x \text{ for all variable } x$$

is a reduction ordering

Prove that

$$f(f(x)) \to f(g(f(x)))$$

is terminating

Building reduction orderings using interpretations

Let $(\mathcal{A}, >)$ with $>$ a well-founded ordering.

Consider a homomorphism τ from ground terms to $(\mathcal{A}, >)$ and let f_τ denote the image of $f \in \mathcal{F}$ using τ; τ and $>$ are constrained to satisfy the monotonicity condition:

$$\forall a, b \in \mathcal{A}, \forall f \in \mathcal{F},\ a > b \text{ implies } f_\tau(\ldots, a, \ldots) > f_\tau(\ldots, b, \ldots).$$

Then the ordering $>_\tau$ defined by:

$$\forall s, t \in \mathcal{T}_\mathcal{F}, s >_\tau t \text{ if } \tau(s) > \tau(t),$$

is well-founded.

Building reduction orderings using interpretations

Then the ordering $>_\tau$ is extended by defining

$$\forall s, t \in \mathcal{T}_{\mathcal{F}}(\mathcal{X}), s >_\tau t \text{ if } \nu(\tau(s)) > \nu(\tau(t))$$

for all assignment ν of values in $\mathcal{A}$ to variables of $\tau(s)$ and $\tau(t)$. Because $>$ is assumed to be well-founded, a rewrite system is terminating if one can find $\mathcal{A}, \tau$ and $>$ as defined above.

Example

Is the reduction induced by $i(f(x, y)) \to f(f(i(x), y), y)$ terminating?

Example

Is the reduction induced by $i(f(x,y)) \to f(f(i(x),y),y)$ terminating?

$$
\begin{aligned}
\tau(i(x)) &= \tau(x)^2 & \qquad \tau(x) &= x \\
\tau(f(x,y)) &= \tau(x) + \tau(y) & \qquad \tau(y) &= y
\end{aligned}
$$

Example

Is the reduction induced by $i(f(x,y)) \to f(f(i(x),y),y)$ terminating?

$$\begin{array}{rcl} \tau(i(x)) & = & \tau(x)^2 \\ \tau(f(x,y)) & = & \tau(x) + \tau(y) \end{array} \qquad \begin{array}{rcl} \tau(x) & = & x \\ \tau(y) & = & y \end{array}$$

Monotonicity: $a > b \text{ implies } f_\tau(a) > f_\tau(b)$
(each function is increasing on natural numbers)

Example

Is the reduction induced by $i(f(x,y)) \to f(f(i(x),y),y)$ terminating?

$$\begin{array}{rcl} \tau(i(x)) & = & \tau(x)^2 \\ \tau(f(x,y)) & = & \tau(x)+\tau(y) \end{array} \qquad \begin{array}{rcl} \tau(x) & = & x \\ \tau(y) & = & y \end{array}$$

Monotonicity: $a > b$ implies $f_\tau(a) > f_\tau(b)$
(each function is increasing on natural numbers)

$$\begin{array}{rcl} \tau(i(f(x,y))) & = & (x+y)^2 = x^2+y^2+2xy \\ \tau(f(f(i(x),y),y)) & = & x^2+2y \end{array}$$

Example

Is the reduction induced by $i(f(x,y)) \to f(f(i(x),y),y)$ terminating?

$$\begin{array}{rcl} \tau(i(x)) & = & \tau(x)^2 \\ \tau(f(x,y)) & = & \tau(x)+\tau(y) \end{array} \qquad \begin{array}{rcl} \tau(x) & = & x \\ \tau(y) & = & y \end{array}$$

Monotonicity: $a > b \text{ implies } f_\tau(a) > f_\tau(b)$
(each function is increasing on natural numbers)

$$\begin{array}{rcl} \tau(i(f(x,y))) & = & (x+y)^2 = x^2+y^2+2xy \\ \tau(f(f(i(x),y),y)) & = & x^2+2y \end{array}$$

For any assignment of positive natural numbers n and m to the variables x and y:

$$n^2+m^2+2nm > n^2+2m$$

Another example

Is the following system terminating?

$$
\begin{array}{rcl}
\ominus \ominus x & \rightarrow & x \\
\ominus(x \oplus y) & \rightarrow & (\ominus x) \oplus (\ominus y) \\
\ominus(x \otimes y) & \rightarrow & (\ominus x) \otimes (\ominus y) \\
x \otimes (y \oplus z) & \rightarrow & (x \otimes y) \oplus (x \otimes z) \\
(x \oplus y) \otimes z & \rightarrow & (x \otimes z) \oplus (y \otimes z)
\end{array}
$$

Another example

Is the following system terminating?

$$\begin{array}{rcl}
\ominus \ominus x & \to & x \\
\ominus(x \oplus y) & \to & (\ominus x) \oplus (\ominus y) \\
\ominus(x \otimes y) & \to & (\ominus x) \otimes (\ominus y) \\
x \otimes (y \oplus z) & \to & (x \otimes y) \oplus (x \otimes z) \\
(x \oplus y) \otimes z & \to & (x \otimes z) \oplus (y \otimes z)
\end{array}$$

Interpretation:

$$\begin{array}{rcl}
\tau(\ominus x) & = & 2^{\tau(x)} \\
\tau(x \oplus y) & = & \tau(x) + \tau(y) + 1 \\
\tau(x \otimes y) & = & \tau(x)\tau(y) \\
\tau(c) & = & 3
\end{array}$$

Exercise

Take the following rewriting system:

$$x + zero \to x$$

$$zero + x \to x$$

$$suc(y) + x \to suc(x + y)$$

$$x + suc(y) \to suc(x + y)$$

Show the termination.

Exercise

Take the following rewriting system:

$$x + zero \to x$$

$$zero + x \to x$$

$$suc(y) + x \to suc(x + y)$$

$$x + suc(y) \to suc(x + y)$$

Show the termination.

$$\tau(x + y) = 2 * (x + y)$$

$$\tau(suc(x)) = x + 1$$

$$\tau(zero) = 1$$

Extensions of reduction ordering

Lexicographical extensions

Let $>$ be an ordering on $\mathcal{T}$.
Its **lexicographical extension** $>^{lex}$ on $\mathcal{T}^n$ is defined as:

$$(s_1, \ldots, s_n) >^{lex} (t_1, \ldots, t_n)$$

if there exists i, $1 \leq i \leq n$ s.t. $s_i >_i t_i$, and $\forall j, 1 \leq j < i$, $s_j = t_j$.

If $>$ is well-founded on $\mathcal{T}$, then $>^{lex}$ is well-founded on $\mathcal{T}^n$.

Lexicographical extensions

Let $>$ be an ordering on $\mathcal{T}$.
Its **lexicographical extension** $>^{lex}$ on $\mathcal{T}^n$ is defined as:

$$(s_1, \ldots, s_n) >^{lex} (t_1, \ldots, t_n)$$

if there exists i, $1 \leq i \leq n$ s.t. $s_i >_i t_i$, and $\forall j, 1 \leq j < i$, $s_j = t_j$.

If $>$ is well-founded on $\mathcal{T}$, then $>^{lex}$ is well-founded on $\mathcal{T}^n$.

FALSE for an infinite product of ordered sets:
$\mathcal{T} = \{a, b\}$ with $a < b$

$$b >^{lex} ab >^{lex} aab >^{lex} aaab >^{lex} \ldots$$

Multiset extensions

Let $>$ an ordering on $\mathcal{T}$.

Its (strict) multiset extension denoted $>^{mult}$ is defined by:

$$\mathcal{M} = \{s_1, \dots, s_m\} >^{mult} \mathcal{N} = \{t_1, \dots, t_n\}$$

if there exist $i \in \{1, \dots, m\}$ and $1 \leq j_1 < \dots < j_k \leq n$ with $k \geq 0$, such that:

- $s_i > t_{j_1}, \dots, s_i > t_{j_k}$ and,
- either $\mathcal{M} - \{s_i\} >^{mult} \mathcal{N} - \{t_{j_1}, \dots, t_{j_k}\}$ or the multisets $\mathcal{M} - \{s_i\}$ and $\mathcal{N} - \{t_{j_1}, \dots, t_{j_k}\}$ are equal.

Multiset extensions - Examples

if $>$ is well-founded on $\mathcal{T}$, then$>^{mult}$ is well-founded on $\mathcal{T}^n$.

Multiset extensions - Examples

if $>$ is well-founded on $\mathcal{T}$, then $>^{mult}$ is well-founded on $\mathcal{T}^n$.

$\{3, 3, 1, 2\} >^{mult} \{3, 1, 2\} >^{mult} \{3\} >^{mult} \{\}$.

Multiset extensions - Examples

if $>$ is well-founded on $\mathcal{T}$, then$>^{mult}$ is well-founded on $\mathcal{T}^n$.

$\{3,3,1,2\} >^{mult} \{3,1,2\} >^{mult} \{3\} >^{mult} \{\}$.
$\{3,3,1,2\} >^{mult} \{3,1,1,2\}$

Multiset extensions - Examples

if $>$ is well-founded on $\mathcal{T}$, then$>^{mult}$ is well-founded on $\mathcal{T}^n$.

$\{3,3,1,2\} >^{mult} \{3,1,2\} >^{mult} \{3\} >^{mult} \{\}$.
$\{3,3,1,2\} >^{mult} \{3,1,1,2\}$
$\{3,3,1,2\} >^{mult} \{3,2,2,2,1,2\}$

Path orderings

Lexicographic Path Ordering (LPO)

For a given precedence $>_{\mathcal{F}}$ on $\mathcal{F}$,

$$s >_{lpo} t$$

if t is a variable, s is not a variable, and $t \in \mathcal{V}ar(s)$

or

$s = f(s_1, .., s_n)$, $t = g(t_1, \ldots, t_m)$, and at least one of the following conditions is satisfied:

Lexicographic Path Ordering (LPO)

For a given precedence $>_{\mathcal{F}}$ on $\mathcal{F}$,

$$s >_{lpo} t$$

if t is a variable, s is not a variable, and $t \in \mathcal{V}ar(s)$
or
$s = f(s_1, .., s_n)$, $t = g(t_1, \ldots, t_m)$, and at least one of the following conditions is satisfied:

1. $f = g$ and $(s_1, \ldots, s_n) >^{lex}_{lpo} (t_1, \ldots, t_m)$ and $\forall j \in \{1, \ldots, m\}, s >_{lpo} t_j$

Lexicographic Path Ordering (LPO)

For a given precedence $>_{\mathcal{F}}$ on $\mathcal{F}$,

$$s >_{lpo} t$$

if t is a variable, s is not a variable, and $t \in \mathcal{V}ar(s)$
or
$s = f(s_1, .., s_n)$, $t = g(t_1, \ldots, t_m)$, and at least one of the following conditions is satisfied:

1. $f = g$ and $(s_1, \ldots, s_n) >_{lpo}^{lex} (t_1, \ldots, t_m)$ and $\forall j \in \{1, \ldots, m\}, s >_{lpo} t_j$
2. $f >_{\mathcal{F}} g$ and $\forall j \in \{1, \ldots, m\}, s >_{lpo} t_j$

Lexicographic Path Ordering (LPO)

For a given precedence $>_{\mathcal{F}}$ on $\mathcal{F}$,

$$s >_{lpo} t$$

if t is a variable, s is not a variable, and $t \in \mathcal{V}ar(s)$
or
$s = f(s_1, .., s_n)$, $t = g(t_1, \ldots, t_m)$, and at least one of the following conditions is satisfied:

1. $f = g$ and $(s_1, \ldots, s_n) >_{lpo}^{lex} (t_1, \ldots, t_m)$ and $\forall j \in \{1, \ldots, m\}, s >_{lpo} t_j$
2. $f >_{\mathcal{F}} g$ and $\forall j \in \{1, \ldots, m\}, s >_{lpo} t_j$
3. $\exists i \in \{1, \ldots, n\}$ s.t either $s_i >_{lpo} t$, or $s_i = t$.

Lexicographic Path Ordering (LPO)

For a given precedence $>_{\mathcal{F}}$ on $\mathcal{F}$,

$$s >_{lpo} t$$

if t is a variable, s is not a variable, and $t \in \mathcal{V}ar(s)$
or
$s = f(s_1, .., s_n)$, $t = g(t_1, \ldots, t_m)$, and at least one of the following conditions is satisfied:

1. $f = g$ and $(s_1, \ldots, s_n) >_{lpo}^{lex} (t_1, \ldots, t_m)$ and $\forall j \in \{1, \ldots, m\}, s >_{lpo} t_j$
2. $f >_{\mathcal{F}} g$ and $\forall j \in \{1, \ldots, m\}, s >_{lpo} t_j$
3. $\exists i \in \{1, \ldots, n\}$ s.t either $s_i >_{lpo} t$, or $s_i = t$.

Theorem
LPO is a simplification ordering

i.e. a reduction ordering that contains the subterm ordering.

Extension of LPO

The definition of the ordering can be extended to terms with variables by adding the following conditions:

1. two different variables are incomparable,
2. a function symbol and a variable are incomparable.

LPO examples

Termination of the Ackermann function:

$$
\begin{array}{rcl}
ack(0, y) & \rightarrow & suc(y) \\
ack(suc(x), 0) & \rightarrow & ack(x, suc(0)) \\
ack(suc(x), suc(y)) & \rightarrow & ack(x, ack(suc(x), y)).
\end{array}
$$

LPO examples

Termination of the Ackermann function:

$$\begin{array}{rcl} ack(0, y) & \rightarrow & suc(y) \\ ack(suc(x), 0) & \rightarrow & ack(x, suc(0)) \\ ack(suc(x), suc(y)) & \rightarrow & ack(x, ack(suc(x), y)). \end{array}$$

With $ack >_{\mathcal{F}} suc$, we can show that

$$\begin{array}{rcl} ack(0, y) & >_{lpo} & suc(y) \\ ack(suc(x), 0) & >_{lpo} & ack(x, suc(0)) \\ ack(suc(x), suc(y)) & >_{lpo} & ack(x, ack(suc(x), y)). \end{array}$$

LPO examples

Termination of the Ackermann function:

$$\begin{array}{rcl} ack(0,y) & \to & suc(y) \\ ack(suc(x),0) & \to & ack(x,suc(0)) \\ ack(suc(x),suc(y)) & \to & ack(x,ack(suc(x),y)). \end{array}$$

With $ack >_{\mathcal{F}} suc$, we can show that

$$\begin{array}{rcl} ack(0,y) & >_{lpo} & suc(y) \\ ack(suc(x),0) & >_{lpo} & ack(x,suc(0)) \\ ack(suc(x),suc(y)) & >_{lpo} & ack(x,ack(suc(x),y)). \end{array}$$

Find a precedence on s and $+$ and show that

$$s(x+y) >_{lpo} s(x)+y$$

Multiset Path Ordering (MPO)

For a given precedence on $\mathcal{F}$,
$s = f(s_1, .., s_n) >_{mpo} t = g(t_1, \ldots, t_m)$ if one at least of the following conditions holds:

1. $f = g$ and $\{s_1, \ldots, s_n\} >_{mpo}^{mult} \{t_1, \ldots, t_m\}$
2. $f >_{\mathcal{F}} g$ and $\forall j \in \{1, \ldots, m\}, s >_{mpo} t_j$
3. $\exists i \in \{1, \ldots, n\}$ such that either $s_i >_{mpo} t$ or $s_i \sim t$
   where $\sim$ means equivalent up to permutation of subterms.

MPO examples

Termination of the *max* function:

$$\begin{array}{rcl} max(n, 0) & \rightarrow & n \\ max(0, n) & \rightarrow & n \\ max(suc(n), suc(m)) & \rightarrow & suc(max(n, m)) \end{array}$$

MPO examples

Termination of the *max* function:

$$\begin{array}{rcl} max(n, 0) & \rightarrow & n \\ max(0, n) & \rightarrow & n \\ max(suc(n), suc(m)) & \rightarrow & suc(max(n, m)) \end{array}$$

Precedence ? $>_{\mathcal{F}}$?

MPO examples

Termination of the *max* function:

$$\begin{aligned} max(n, 0) &\rightarrow n \\ max(0, n) &\rightarrow n \\ max(suc(n), suc(m)) &\rightarrow suc(max(n, m)) \end{aligned}$$

Precedence $max >_{\mathcal{F}} suc$

Show that

$$-(1 \times (1 + 0)) >_{mpo} (-1) + (-(0 + 1))$$

Recursive Path Ordering with Status (RPO)

For a given precedence on $\mathcal{F}$,
$s = f(s_1, .., s_n) >_{rpos} t = g(t_1, \ldots, t_m)$ if one at least of the following condition holds:

1. $f >_{\mathcal{F}} g$ and $\forall j \in \{1, \ldots, m\}, s >_{rpos} t_j$
2. $f >_{\mathcal{F}} g$ and $\exists i \in \{1, \ldots, n\}$ such that either $s_i >_{rpos} t$ or $s_i \sim t$.
3. $f = g$, $Stat(f) = lex$, $(s_1, \ldots, s_n) >^{lex}_{rpos} (t_1, \ldots, t_m)$, and $\forall k \in \{1, \ldots, n\}, s >_{rpos} t_k$.
4. $f = g$, $Stat(f) = mult$ and $\{s_1, \ldots, s_n\} >^{mult}_{rpos} \{t_1, \ldots, t_m\}$.

Exercises

Words on $\mathbb{N}$

$$u.(i+1).v \to u.i.i.v$$

Example : $12.4.6.7 \to 12.3.3.6.7 \to 12.3.3.5.5.7 \ldots$

$$\begin{aligned} ack(0,y) &\to y+1 \\ ack(x+1,0) &\to ack(x,1) \\ ack(x+1,y+1) &\to ack(x, ack(x+1,y)). \end{aligned}$$

Show the termination without using a LPO

$$(x+y)+z \to x+(y+z)$$

$$x+(y+z) \to (x+y)+z$$

Words on $\mathbb{N}$

$$u.(i+1).v \to u.i.i.v$$

Example : $12.4.6.7 \to 12.3.3.6.7 \to 12.3.3.5.5.7 \ldots$

Multiset ordering. Not a lexicographic ordering

$$\begin{array}{rcl} ack(0, y) & \to & y+1 \\ ack(x+1, 0) & \to & ack(x, 1) \\ ack(x+1, y+1) & \to & ack(x, ack(x+1, y)). \end{array}$$

Show the termination without using a LPO

Lexicographic ordering

$$(x+y)+z \to x+(y+z)$$

LPO (left to right)

$$x+(y+z) \to (x+y)+z$$

LPO (right to left)

Take the following rewriting system:

$$x + zero \to x$$

$$zero + x \to x$$

$$suc(y) + x \to suc(y + x)$$

$$x + suc(y) \to suc(x + y)$$

Show the termination without using an interpretation

Take the following rewriting system:

$$x + zero \to x$$

$$zero + x \to x$$

$$suc(y) + x \to suc(y + x)$$

$$x + suc(y) \to suc(x + y)$$

Show the termination without using an interpretation

LPO or MPO with $+ > suc$

Show the termination of the following rewriting system ($+$ is associative and commutative):

$$x + zero \rightarrow x$$

$$x * zero \rightarrow zero$$

$$suc(y) + x \rightarrow suc(x + y)$$

$$x * suc(y) \rightarrow x + (x * y)$$

Show the termination of the following rewriting system ($+$ is associative and commutative):

$$x + zero \to x$$
$$x * zero \to zero$$
$$suc(y) + x \to suc(x + y)$$
$$x * suc(y) \to x + (x * y)$$

$\tau(x * y) = 2^{x*y}$
$\tau(x + y) = x * y$
$\tau(suc(x)) = x + 1$
$\tau(zero) = 1$

or LPO or MPO with $* > + > suc$

Syntactic filtering

$$
\begin{array}{lll}
 & t \ll^? t \wedge P & \\
\to & P & \\
 & f(t_1, \ldots, t_n) \ll^? f(t'_1, \ldots, t'_n) \wedge P & \\
\to & \bigwedge_{i=1,\ldots,n} t_i \ll^? t'_i \wedge P & \\
 & f(t_1, \ldots, t_n) \ll^? g(t'_1, \ldots, t'_m) \wedge P & \\
\to & \mathcal{F} & \text{if } f \neq g \\
 & x \ll^? t \wedge x \ll^? t' \wedge P & \\
\to & \mathcal{F} & \text{if } t \neq t' \\
 & f(t_1, \ldots, t_n) \ll^? x \wedge P & \\
\to & \mathcal{F} & \text{if } x \in \mathcal{X}
\end{array}
$$

Termination?

Groups

$$
\begin{array}{lcl}
x + e & \rightarrow & x \\
e + x & \rightarrow & x \\
x + (y + z) & \rightarrow & (x + y) + z \\
x + i(x) & \rightarrow & e \\
i(x) + x & \rightarrow & e
\end{array}
\qquad
\begin{array}{lcl}
i(e) & \rightarrow & e \\
(y + i(x)) + x & \rightarrow & y \\
(y + x) + i(x) & \rightarrow & y \\
i(i(x)) & \rightarrow & x \\
i(x + y) & \rightarrow & i(y) + i(x)
\end{array}
$$

Termination:

Groups

$$\begin{array}{lcl} x + e & \rightarrow & x \\ e + x & \rightarrow & x \\ x + (y + z) & \rightarrow & (x + y) + z \\ x + i(x) & \rightarrow & e \\ i(x) + x & \rightarrow & e \end{array} \qquad \begin{array}{lcl} i(e) & \rightarrow & e \\ (y + i(x)) + x & \rightarrow & y \\ (y + x) + i(x) & \rightarrow & y \\ i(i(x)) & \rightarrow & x \\ i(x + y) & \rightarrow & i(y) + i(x) \end{array}$$

Termination:
$\tau(x + y) = x + 2 * y$
$\tau(i(x)) = x^2$
$\tau(e) = 1$

or LPO (right to left) with $i > + > e$