On Non-Looping Term Rewriting & Proving Non-Looping Non-Termination Automatically

Astrid van der Jagt

In this presentation

Non-Looping Non-Termination of TRS

- Term Rewrite Systems (TRS)
- Paper 1: On Non-Looping Term Rewriting
  - Loops/Non-Loops
  - Inner-looping TRS
  - Normal TRS
- Paper 2: Proving Non-Looping Non-Termination Automatically
  - Pattern terms and pattern rules
  - Proving Non-Termination

Term Rewrite Systems (TRS) (1/7)

- Terms
  - Signature *F* function symbols + arity
  - Infinite set of variables *V*
- Rules
- Context
- Substitution

Term Rewrite Systems (TRS) (2/7)

Terms and Rules

- Terms:
  - F = { plus (2), s (1), 0 (0) }
  - x, y $\in V$

- Rules:
  - plus(x, 0) $\rightarrow$ x
  - plus(x, s(y)) $\rightarrow$ plus(s(x), y)

- $s \rightarrow_R t$ if there exists a rewrite rule $l \rightarrow r$, a substitution σ and a context C such that $s = C[l\sigma]$ and $t = C[r\sigma]$

Term Rewrite Systems (TRS) (3/7)

Context

- $s \to_R t$ if there exists a rewrite rule $l \to r$, a substitution σ and a context C such that $s = C[l\sigma]$ and $t = C[r\sigma]$

- Let C be a context with a hole ☐
  Then $C[t]$ is the term obtained from replacing ☐ with a term t.

- Example:
  - C = f(x, g(☐, z))
  - $C[t]$ = f(x, g(t, z))

Term Rewrite Systems (TRS) (4/7)

Substitution

- $s \rightarrow_R t$ if there exists a rewrite rule $l \rightarrow r$, a substitution σ and a context C such that $s = C[l\sigma]$ and $t = C[r\sigma]$

- A substitution is a mapping σ from V to the Terms.

  $\sigma = [x \mapsto s(s(0)),\ y \mapsto s(0)]$

- Applying σ to a term t = plus(x, s(y)), we get

  t σ = plus(s(s(0)), s(s(0)))

Term Rewrite Systems (TRS) (5/7)

Rewriting

- $s \rightarrow_R t$ if there exists a rewrite rule $l \rightarrow r$, a substitution σ and a context C such that $s = C[l\sigma]$ and $t = C[r\sigma]$

- Rules:
  - plus(x, 0) → x
  - plus(x, s(y)) → plus(s(x), y)

$C = \square$

$\sigma = [x \mapsto s(s(0)), y \mapsto s(0)]$

- plus(s(s(0)), s(s(0))) $\rightarrow_R$ plus(s(s(s(0))), s(0))
  $\rightarrow_R$ plus(s(s(s(s(0)))), 0)
  $\rightarrow_R$ s(s(s(s(0))))

Term Rewrite Systems (TRS) (6/7)

Non-Termination

- Rules:
  - plus(x, 0) $\to$ x
  - plus(x, s(y)) $\to$ plus(y, s(x))

- plus(s(s(0)), s(s(0))) $\to_R$ plus(s(0), s(s(s(0)))) $\to_R$ plus(s(s(0)), s(s(0))) $\to_R \cdots$

Term Rewrite Systems (TRS) (7/7)

Termination

- A TRS is terminating if it does not admit any infinite reduction sequences

On Non-Looping Term Rewriting

Yi Wang and Masahiko Sakai (2006)

Loops

Looping TRS

A reduction sequence loops if it contains $t \to_R^+ C[t\sigma]$. A TRS admits a loop if there is a looping reduction sequence.

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

$t = f(x)$

$C = h(\square)$

$\sigma = [x \mapsto g(x)]$

$$
\begin{aligned}
f(x) &\to_R h(f(g(x))) \\
&\to_R h(h(f(g(g(x))))) \\
&\to_R h(h(h(f(g(g(g(x))))))) \\
&\to_R \cdots
\end{aligned}
$$

Non-Loops (1/5)

Non-Looping TRS

A rewrite sequence is non-looping if it is infinite and does not contain any loop. A TRS is non-looping if it admits an non-looping sequence. A TRS is properly non-looping if it is non-looping and does not admit any looping sequence.

$b(c) \rightarrow d(c)$
$b(d(x)) \rightarrow d(b(x))$
$a(d(x)) \rightarrow a(b(b(x)))$

$a(b(c)) \rightarrow_R^2 a(b(b(c)))$
$\rightarrow_R^3 a(b(b(b(c))))$
$\rightarrow_R^4 \dots$

Non-Loops (2/5)

Inner-Looping sequence/TRS

Given a TRS R, let t be a term, an *inner-looping sequence* is of the form:

$$C[\Delta^{\ell_1} t \delta^{\ell_1}] \quad \rightarrow_R^+ \quad C[\Delta^{\ell_2} t \delta^{\ell_2}] \quad \rightarrow_R^+ \quad \dots$$

Where C and Δ are contexts, δ is a substitution, $\{\ell_i\}$ is an infinite sequence of natural numbers.

A TRS R is *inner-looping* if R admits an inner-looping sequence. A TRS R is *properly inner-looping* if R is inner-looping and does not admit any looping sequence.

Non-Loops (3/5)

Inner-Looping TRS

$C[\Delta^{\ell_1} t\delta^{\ell_1}] \quad \to_R^+ \quad C[\Delta^{\ell_2} t\delta^{\ell_2}] \quad \to_R^+ \quad \dots$ for term t,

where C and Δ are contexts, δ is a substitution, $\{\ell_i\}$ is an infinite sequence.

$b(c) \to d(c)$
$b(d(x)) \to d(b(x))$
$a(d(x)) \to a(b(b(x)))$

$a(b(c)) \to_R^2 a(b(b(c)))$
$\to_R^3 a(b(b(b(c))))$
$\to_R^4 \dots$

$t = c$, $C = a(\square)$, $\Delta = b(\square)$, $\delta = \varnothing$, $\ell_i = i$

Non-Loops (4/5)

Normal TRS

Are there non-looping rewrite sequences without any patterns?

Normal numbers: real numbers whose digits show a random distribution with all digits appearing equally.

Normal sequence: infinite reduction sequence with all function symbols appearing equally in every term of the sequence.

Normal TRS is a TRS that admits a normal sequence.

Non-Loops (5/5)

The inner-looping property and the properly inner-looping property for TRSs are undecidable.

The non-looping property is undecidable.

Conclusion

On Non-Looping Term Rewriting

- Looping TRS & Non-looping TRS
- Inner-looping TRS
- Normal TRS
- Undecidability

Proving Non-Looping Non-Termination Automatically

Fabian Emmes, Tim Enger and Jürgen Giesl (2012)

Proving Non-Looping Non-Termination Automatically

- Pattern terms
- Pattern rules
- The technique: narrowing

Our example (1/2)

Proving non-looping non-termination automatically

```
def f():
  while (gt(x, y)):
    x    =    dbl(x)
    y    =    y + 1
```

Our example (2/2)

Proving non-looping non-termination automatically

f(tt, x, y)	→ f(gt(x, y), dbl(x), s(y))
gt(s(x), 0)	→ tt
gt(0, y)	→ ff
gt(s(x), s(y))	→ gt(x, y)
dbl(x)	→ times(s(s(0)), x)
times(x, 0)	→ 0
times(x, s(y))	→ plus(times(x, y), x)
plus(x, 0)	→ x
plus(x, s(y))	→ plus(s(x), y)

```
def f():
  while (gt(x, y)):
    x   =   dbl(x)
    y   =   y + 1
```

Pattern Terms (1/2)

Describing a set of terms

A mapping $n \mapsto t\, \sigma^n\, \mu$

- t in Terms
- σ, μ are substitutions

$$gt(s(x), 0) \to tt$$
$$gt(s(x), s(y)) \to gt(x, y)$$

Base term	Pumping substitution	Closing substitution
$gt(s(x), s(y))$	$[x \mapsto s(x), y \mapsto s(y)]^n$	$[x \mapsto s(x), y \mapsto 0]$

Pattern Terms (2/2)

Interpreting a pattern term

A mapping $n \mapsto t\ \sigma^n\ \mu$

For n = 1, we apply the pumping substitution once to the base term, and then apply the closing substitution: gt(s(s(x)), s(s(y))) gt(s(s(s(x))), s(s(0)))

Base term	Pumping substitution	Closing substitution
gt(s(x), s(y))	$[x \mapsto s(x), y \mapsto s(y)]^n$	$[x \mapsto s(x), y \mapsto 0]$

Pattern Rules

Describing a set of rewrite sequences

$gt(s(x), 0) \to tt$

$gt(s(x), s(y)) \to gt(x, y)$

A pattern rule $p \hookrightarrow q$ is correct w.r.t. a TRS R if $p(n) \to_R^+ q(n)$ for all $n \in \mathbb{N}$.

$gt(s^{n+1}(x), s^{n+1}(y))$ $\hookrightarrow$ $gt(s^{n+2}(x), s^{n+1}(0))$ and it holds that $gt(s^{n+2}(x), s^{n+1}(0)) \to_R^+ tt$

Base term	Pumping substitution	Closing substitution
$gt(s(x), s(y))$	$[x \mapsto s(x), y \mapsto s(y)]^n$	$[x \mapsto s(x), y \mapsto 0]$
	$\hookrightarrow$	
tt	$\varnothing^n$	$\varnothing$

Base term	Pumping substitution	Closing substitution
$gt(s(x), s(y))$	$[x \mapsto s(x), y \mapsto s(y)]^n$	$[x \mapsto s(x), y \mapsto 0]$
	$\hookrightarrow$	
tt	$\varnothing^n$	$\varnothing$

Proving Non-Looping Non-Termination Automatically

The technique

1. Generate pattern rules from the TRS.
2. Modify to avoid empty pattern substitutions.
3. Prepare for narrowing p ↪ q with p’ ↪ q’:
   a. Make the base term of q at some position equal to the base term of p’.
   b. Make the pumping substitutions and closing substitutions of the two pattern rules equal.
4. Narrow p ↪ q with p’ ↪ q’.
5. Check for non-termination of the resulting rule s ↪ t: is t a specialization of s?

Detecting Non-Termination

$f(tt, x, y) \to f(gt(x, y), dbl(x), s(y))$

Theorem

Let $s\,\sigma^n\,\mu \hookrightarrow t\,(\sigma^m\,\theta)^n\,(\mu\,v)$ be correct w.r.t. a TRS R, where θ commutes with both σ and μ. If there is a position π and some $b \in \mathbb{N}$ such that $s\,\sigma^b = t|_\pi$, then R is non-terminating.

$s\,\sigma^n\,\mu \to_R^+ t\,(\sigma^m\,\theta)^n\,(\mu\,v)$ and if we zoom in on the subterm where $s\,\sigma^b = t|_\pi$ we get

$$t|_\pi\,(\sigma^m\,\theta)^n\,(\mu\,v) = s\,\sigma^b\,(\sigma^m\,\theta)^n\,(\mu\,v)$$

$$= s\,\sigma^{mn+b}\,\mu\,\theta^n\,v$$

Creating Pattern Rules (1/16)

Using *inference rules*

$\mathrm{gt}(s(x), s(y)) \rightarrow \mathrm{gt}(x, y)$

$$\frac{}{s\ \varnothing^n\ \varnothing \hookrightarrow t\ \varnothing^n\ \varnothing} \quad \text{if } s \rightarrow t \in R$$

Base term	Pumping substitution	Closing substitution
gt(s(x), s(y))	$\varnothing^n$	$\varnothing$
	$\hookrightarrow$	
gt(x, y)	$\varnothing^n$	$\varnothing$

Creating Pattern Rules (2/16)

Using *inference rules*

gt(s(x), s(y)) $\rightarrow$ gt(x, y)

$$\frac{s\ \varnothing^n\ \varnothing \hookrightarrow t\ \varnothing^n\ \varnothing}{s\ \sigma^n\ \varnothing \hookrightarrow t\ \theta^n\ \varnothing}$$

if $s\theta = t\sigma$, and

θ *commutes* with σ

Base term	Pumping substitution	Closing substitution
gt(s(x), s(y))	$[x \mapsto s(x), y \mapsto s(y)]^n$	$\varnothing$
	$\hookrightarrow$	
gt(x, y)	$\varnothing^n$	$\varnothing$

Creating Pattern Rules (3/16)

$gt(s(x), s(y)) \rightarrow gt(x, y)$

Equivalence by *domain renaming*

$$\frac{p \hookrightarrow q}{p' \hookrightarrow q'}$$ if p is equivalent to p' and q is equivalent to q'

Base term	Pumping substitution	Closing substitution
$gt(s(x'), s(y'))$	$[x' \mapsto s(x'), y' \mapsto s(y')]^n$	$[x' \mapsto x, y' \mapsto y]$
	$\hookrightarrow$	
$gt(x, y)$	$\varnothing^n$	$\varnothing$

Creating Pattern Rules (4/16)

$gt(s(x), s(y)) \to gt(x, y)$

Equivalence by *irrelevant pattern substitutions*

$$\frac{p \hookrightarrow q}{p' \hookrightarrow q'}$$ if p is equivalent to p' and q is equivalent to q'

Base term	Pumping substitution	Closing substitution
$gt(s(x'), s(y'))$	$[x' \mapsto s(x'), y' \mapsto s(y')]^n$	$[x' \mapsto x, y' \mapsto y]$

$\hookrightarrow$

Base term	Pumping substitution	Closing substitution
$gt(x, y)$	$[x' \mapsto s(x'), y' \mapsto s(y')]^n$	$[x' \mapsto x, y' \mapsto y]$

Creating Pattern Rules (5/16)

Instantiation

$gt(s(x), 0) \to tt$

$gt(s(x), s(y)) \to gt(x, y)$

We instantiate x to be s(x) and y to be 0.

Base term	Pumping substitution	Closing substitution
$gt(s(x'), s(y'))$	$[x' \mapsto s(x'), y' \mapsto s(y')]^n$	$[x' \mapsto s(x), y' \mapsto 0]$

$\hookrightarrow$

$gt(s(x), 0)$	$[x' \mapsto s(x'), y' \mapsto s(y')]^n$	$[x' \mapsto s(x), y' \mapsto 0]$

Creating Pattern Rules (6/16)

Pattern rule from TRS

$gt(s(x), 0) \to tt$

$gt(s(x), s(y)) \to gt(x, y)$

$$\frac{}{s\ \varnothing^n\ \varnothing \hookrightarrow t\ \varnothing^n\ \varnothing} \quad \text{if } s \to t \in R$$

Base term	Pumping substitution	Closing substitution
$gt(s(x), 0)$	$\varnothing^n$	$\varnothing$
	$\hookrightarrow$	
tt	$\varnothing^n$	$\varnothing$

Creating Pattern Rules (7/16)

Equivalence by *irrelevant pattern substitutions*

$gt(s(x), 0) \rightarrow tt$

$gt(s(x), s(y)) \rightarrow gt(x, y)$

$$\frac{p \hookrightarrow q}{p' \hookrightarrow q'}$$ if p is equivalent to p' and q is equivalent to q'

Since x' and y' are not relevant in the pattern rule:

Base term	Pumping substitution	Closing substitution
$gt(s(x), 0)$	$[x' \mapsto s(x'), y' \mapsto s(y')]^n$	$[x' \mapsto s(x), y' \mapsto 0]$
	$\hookrightarrow$	
tt	$[x' \mapsto s(x'), y' \mapsto s(y')]^n$	$[x' \mapsto s(x), y' \mapsto 0]$

Creating Pattern Rules (8/16)

Narrowing

$gt(s(x), 0) \to tt$
$gt(s(x), s(y)) \to gt(x, y)$

Base term	Pumping substitution	Closing substitution
$gt(s(x'), s(y'))$	$[x' \mapsto s(x'), y' \mapsto s(y')]^n$	$[x' \mapsto s(x), y' \mapsto 0]$

$\hookrightarrow$

$gt(s(x), 0)$	$[x' \mapsto s(x'), y' \mapsto s(y')]^n$	$[x' \mapsto s(x), y' \mapsto 0]$

Base term	Pumping substitution	Closing substitution
$gt(s(x), 0)$	$[x' \mapsto s(x'), y' \mapsto s(y')]^n$	$[x' \mapsto s(x), y' \mapsto 0]$

$\hookrightarrow$

tt	$[x' \mapsto s(x'), y' \mapsto s(y')]^n$	$[x' \mapsto s(x), y' \mapsto 0]$

Creating Pattern Rules (9/16)

Narrowing

$$\frac{s\,\sigma^n\,\mu \hookrightarrow t\,\sigma^n\,\mu \qquad u\,\sigma^n\,\mu \hookrightarrow v\,\sigma^n\,\mu}{s\,\sigma^n\,\mu \hookrightarrow t[v]_\pi\,\sigma^n\,\mu} \quad \text{if } t\,|_\pi = u$$

Base term	Pumping substitution	Closing substitution
$gt(s(x'), s(y'))$	$[x' \mapsto s(x'), y' \mapsto s(y')]^n$	$[x' \mapsto s(x), y' \mapsto 0]$
	$\hookrightarrow$	
tt	$[x' \mapsto s(x'), y' \mapsto s(y')]^n$	$[x' \mapsto s(x), y' \mapsto 0]$

Creating Pattern Rules (10/16)

$f(tt, x, y) \to f(gt(x, y), dbl(x), s(y))$

Pattern rule from TRS

$$\frac{}{s\,\varnothing^n\,\varnothing \hookrightarrow t\,\varnothing^n\,\varnothing} \quad \text{if } s \to t \in R$$

Base term	Pumping substitution	Closing substitution
$f(tt, x, y)$	$\varnothing^n$	$\varnothing$
	$\hookrightarrow$	
$f(gt(x, y), dbl(x), s(y))$	$\varnothing^n$	$\varnothing$

Creating Pattern Rules (11/16)

Instantiation

$f(tt, x, y) \to f(gt(x, y), dbl(x), s(y))$

We instantiate x to be s(x’) and y to be s(y’).

Base term	Pumping substitution	Closing substitution
$f(tt, s(x'), s(y'))$	$\varnothing^n$	$\varnothing$
	$\hookrightarrow$	
$f(gt(s(x'), s(y')), dbl(s(x')), s^2(y'))$	$\varnothing^n$	$\varnothing$

Creating Pattern Rules (12/16)

$f(tt, x, y) \to f(gt(x, y), dbl(x), s(y))$

Instantiation of pumping substitutions

$$\frac{s\ \sigma^n\ \mu \hookrightarrow t\ \theta^n\ v}{s\ (\sigma\rho)^n\ \mu \hookrightarrow t\ (\theta\rho)^n\ v}$$

if ρ commutes with σ, μ, θ, v

Base term	Pumping substitution	Closing substitution
$f(tt, s(x'), s(y'))$	$[x' \mapsto s(x'), y' \mapsto s(y')]^n$	$\varnothing$
	$\hookrightarrow$	
$f(gt(s(x'), s(y')), dbl(s(x')), s^2(y'))$	$[x' \mapsto s(x'), y' \mapsto s(y')]^n$	$\varnothing$

Creating Pattern Rules (13/16)

$f(tt, x, y) \rightarrow f(gt(x, y), dbl(x), s(y))$

Instantiation of closing substitutions

$$\frac{s\ \sigma^n\ \mu \hookrightarrow t\ \theta^n\ \nu}{s\ \sigma^n\ (\mu\rho) \hookrightarrow t\ \theta^n\ (\nu\rho)}$$

Base term	Pumping substitution	Closing substitution
$f(tt, s(x'), s(y'))$	$[x' \mapsto s(x'), y' \mapsto s(y')]^n$	$[x' \mapsto s(x), y' \mapsto 0]$

$\hookrightarrow$

$f(gt(s(x'), s(y')), dbl(s(x')), s^2(y'))$ $\quad [x' \mapsto s(x'), y' \mapsto s(y')]^n \quad [x' \mapsto s(x), y' \mapsto 0]$

Creating Pattern Rules (14/16)

$f(tt, x, y) \to f(gt(x, y), dbl(x), s(y))$

Narrowing

Base term	Pumping substitution	Closing substitution
$f(tt, s(x'), s(y'))$	$[x' \mapsto s(x'), y' \mapsto s(y')]^n$	$[x' \mapsto s(x), y' \mapsto 0]$

$\hookrightarrow$

$f(gt(s(x'), s(y')), dbl(s(x')), s^2(y'))$ $[x' \mapsto s(x'), y' \mapsto s(y')]^n$ $[x' \mapsto s(x), y' \mapsto 0]$

Base term	Pumping substitution	Closing substitution
$gt(s(x'), s(y'))$	$[x' \mapsto s(x'), y' \mapsto s(y')]^n$	$[x' \mapsto s(x), y' \mapsto 0]$

$\hookrightarrow$

tt	$[x' \mapsto s(x'), y' \mapsto s(y')]^n$	$[x' \mapsto s(x), y' \mapsto 0]$

Creating Pattern Rules (15/16)

Narrowing

$f(tt, x, y) \rightarrow f(gt(x, y), dbl(x), s(y))$

$dbl(x) \rightarrow times(s(s(0)), x)$

$$\frac{s\ \sigma^n\ \mu \hookrightarrow t\ \sigma^n\ \mu \qquad u\ \sigma^n\ \mu \hookrightarrow v\ \sigma^n\ \mu}{s\ \sigma^n\ \mu \hookrightarrow t[v]_\pi\ \sigma^n\ \mu} \quad \text{if } t\,|_\pi = u$$

Base term	Pumping substitution	Closing substitution
$f(tt, s(x'), s(y'))$	$[x' \mapsto s(x'), y' \mapsto s(y')]^n$	$[x' \mapsto s(x), y' \mapsto 0]$
$\hookrightarrow$		
$f(tt, dbl(s(x')), s^2(y'))$	$[x' \mapsto s(x'), y' \mapsto s(y')]^n$	$[x' \mapsto s(x), y' \mapsto 0]$

Creating Pattern Rules (16/16)

Rewriting

$f(tt, x, y) \to f(gt(x, y), dbl(x), s(y))$

$dbl(x) \to times(s(s(0)), x)$

For rewriting, we can just rewrite the right-hand side of the pattern rule with the rewriting rules from the original TRS.

Base term	Pumping substitution	Closing substitution
$f(tt, s(x'), s(y'))$	$[x' \mapsto s(x'), y' \mapsto s(y')]^n$	$[x' \mapsto s(x), y' \mapsto 0]$

$\hookrightarrow$

$f(tt, times(s^2(0), s(x')), s^2(y'))$ $[x' \mapsto s(x'), y' \mapsto s(y')]^n$ $[x' \mapsto s(x), y' \mapsto 0]$

Creating Pattern Rules (16a/16c)

times(x, s(y))→plus(times(x, y), x)

Pattern rule from TRS

$$\frac{}{s \varnothing^n \varnothing \hookrightarrow t \varnothing^n \varnothing} \quad \text{if } s \to t \in R$$

Base term	Pumping substitution	Closing substitution
times(x, s(y))	$\varnothing^n$	$\varnothing$
$\hookrightarrow$		
plus(times(x, y), x)	$\varnothing^n$	$\varnothing$

Creating Pattern Rules (16b/16c)

times(x, s(y))→plus(times(x, y), x)

$$\frac{s\ \varnothing^n\ \varnothing\ \hookrightarrow\ t\ \varnothing^n\ \varnothing}{s\ \sigma^n\ \varnothing\ \hookrightarrow\ t[z]_\pi\ (\sigma\ \cup\ [z \mapsto t[z]_\pi])^n\ [z \mapsto t|_\pi]}$$

if $s = t\,|_\pi \sigma$ and

$z \in V$ is fresh

Base term	Pumping substitution	Closing substitution
times(x, s(y))	$[y \mapsto s(y)]^n$	$\varnothing$
	$\hookrightarrow$	
plus(times(x, y), x)	$[y \mapsto s(y), z \mapsto plus(z, x)]^n$	$[z \mapsto times(x, y)]$

Creating Pattern Rules (16c/16c)

$f(tt, x, y) \to f(gt(x, y), dbl(x), s(y))$

Some steps further

After also narrowing with times and cleaning the pattern rule:

Base term	Pumping substitution	Closing substitution
$f(tt, s(x'), s(y'))$	$[x' \mapsto s(x'), y' \mapsto s(y')]^n$	$[x' \mapsto s(x'), y' \mapsto 0]$
	$\hookrightarrow$	
$f(tt, s^2(x'), s^2(y'))$	$[x' \mapsto s^2(x'), y' \mapsto s(y')]^n$	$[x' \mapsto times(s^2(0), s(x')), y' \mapsto 0]$

Detecting Non-Termination (1/4)

$f(tt, x, y) \to f(gt(x, y), dbl(x), s(y))$

Some steps further

After also narrowing with times and cleaning the pattern rule.

Check whether the pattern substitutions of the right-hand side are specializations of the pattern substitutions of the left-hand side

Base term	Pumping substitution	Closing substitution
$f(tt, s^2(x'), s(y'))$	$[x' \mapsto s(x'), y' \mapsto s(y')]^n$	$[y' \mapsto 0]$
	$\hookrightarrow$	
$f(tt, s^3(x'), s^2(y'))$	$[x' \mapsto s^2(x'), y' \mapsto s(y')]^n$	$[x' \mapsto s(times(s^2(0), x')), y' \mapsto 0]$

Detecting Non-Termination (2/4)

$f(tt, x, y) \to f(gt(x, y), dbl(x), s(y))$

Theorem

Let $s\, \sigma^n\, \mu \hookrightarrow t\, (\sigma^m\, \theta)^n\, (\mu\, v)$ be correct w.r.t. a TRS R, where θ commutes with both σ and μ. If there is a position π and some $b \in \mathbb{N}$ such that $s\, \sigma^b = t|_\pi$, then R is non-terminating.

$s\, \sigma^n\, \mu \to_R^+ t\, (\sigma^m\, \theta)^n\, (\mu\, v)$ and if we zoom in on the subterm where $s\, \sigma^b = t|_\pi$ we get

$$t|_\pi\, (\sigma^m\, \theta)^n\, (\mu\, v) = s\, \sigma^b\, (\sigma^m\, \theta)^n\, (\mu\, v)$$

$$= s\, \sigma^{mn+b}\, \mu\, \theta^n\, v$$

Detecting Non-Termination (3/4)

$f(tt, x, y) \to f(gt(x, y), dbl(x), s(y))$

Theorem

Let $s\ \sigma^n\ \mu\ \hookrightarrow\ t\ (\sigma^m\ \theta)^n\ \mu\ \nu$ be correct w.r.t. a TRS R, where θ commutes with both σ and μ. If there is a position π and some $b \in \mathbb{N}$ such that $s\ \sigma^b = t|_\pi$, then R is non-terminating.

Base term	Pumping substitution	Closing substitution
$f(tt, s^2(x'), s(y'))$	$[x' \mapsto s(x'), y' \mapsto s(y')]^n$	$[y' \mapsto 0]$

$\hookrightarrow$

$f(tt, s^3(x'), s^2(y'))$	$[x' \mapsto s^2(x'), y' \mapsto s(y')]^n$	$[x' \mapsto s(times(s^2(0), x')), y' \mapsto 0]$

Detecting Non-Termination (4/4)

$f(tt, x, y) \to f(gt(x, y), dbl(x), s(y))$

Theorem

Let $s\ \sigma^n\ \mu \hookrightarrow t\ (\sigma^m\ \theta)^n\ \mu\ v$ be correct w.r.t. a TRS R, where θ commutes with both σ and μ. If there is a position π and some $b \in \mathbb{N}$ such that $s\ \sigma^b = t|_\pi$, then R is non-terminating. And since $s\ \sigma = f(tt, s^3(x'), s^2(y')) = f(tt, s^3(x'), s^2(y'))$, there is an infinite reduction sequence and the TRS is non-terminating.

Base term	Pumping substitution	Closing substitution
$f(tt, s^2(x'), s(y'))$	$[x' \mapsto s(x'), y' \mapsto s(y')]^n$	$[y' \mapsto 0]$

$\hookrightarrow$

$f(tt, s^3(x'), s^2(y'))\ ([x' \mapsto s(x'), y' \mapsto s(y')][x' \mapsto s(x')])^n\ ([y' \mapsto 0][x' \mapsto s(times(s^2(0), x'))])$

Proving Non-Looping Non-Termination Automatically

Recap of the technique

1. Generate pattern rules from the TRS.
2. Modify to avoid empty pattern substitutions.
3. Prepare for narrowing $p \hookrightarrow q$ with $p' \hookrightarrow q'$:
   a. Make the base term of q at some position equal to the base term of p’.
   b. Make the pumping substitutions and closing substitutions of the two pattern rules equal.
4. Narrow $p \hookrightarrow q$ with $p' \hookrightarrow q'$.
5. Check for non-termination of the resulting rule $s \hookrightarrow t$: is t a specialization of s?

How is this automated?

Soundness of the inference rules

For all the inference rules of the form

$$\frac{p_1 \hookrightarrow q_1 \qquad \dots \qquad p_k \hookrightarrow q_k}{p \hookrightarrow q}$$

If all pattern rules $p_1 \hookrightarrow q_1, \dots, p_k \hookrightarrow q_k$ are correct w.r.t. a TRS R, then the pattern rule $p \hookrightarrow q$ is also correct w.r.t. R.

Final remarks

Proving Non-Looping Non-Termination Automatically

- The technique can prove non-termination of looping and non-looping TRS
- Tested on 58 typical non-looping non-terminating TRSs, and were able to prove non-termination in 75.9% of the non-looping examples
- There are TRSs for which the non-termination cannot be proved with the technique.