Termination Tools

Nao Hirokawa

JAIST

14th ISR, August 27, 2024

https://www.jaist.ac.jp/~hirokawa/24isr/

Contents: Proving (Non-)Termination Automatically

TRS category in termCOMP 2024:

	tool/method	YES	NO
	AProVE	1030	282
	NaTT	876	169
	MU-TERM	717	135
	NTI	293	275
	autonon	—	233
1	**linear interpretations**	145	—
2	**dependency pair method with linear and max/plus**	431	—
3	**termination of imperative programs**	—	—
4	**loop detection**	—	$\geqslant 178$

Termination by Linear Interpretations

Well-Founded Monotone Algebras

Definition

let $\mathcal{A} = (A, \{f_{\mathcal{A}}\})$ be algebra equipped with strict order $>$

- $\mathcal{A}$ is well-founded if $>$ is well-founded
- $\mathcal{A}$ is (strictly) monotone if

$$a_i > b \implies f_{\mathcal{A}}(a_1, \dots, a_i, \dots, a_n) > f_{\mathcal{A}}(a_1, \dots, b, \dots, a_n)$$

  for all $f^{(n)} \in \mathcal{F}$, $a_1, \dots, a_n, b \in A$, and $i \in \{1, \dots, n\}$
- $s >_{\mathcal{A}} t$ if $[\alpha]_{\mathcal{A}}(s) > [\alpha]_{\mathcal{A}}(t)$ for all assignments $\alpha : \mathcal{V} \to A$

Theorem (Lankford 1979, Zantema 1994)

$>_{\mathcal{A}}$ is reduction order if $\mathcal{A}$ is well-founded monotone algebra

Termination Proof of Addition

consider TRS $\mathcal{R}$ and well-founded monotone algebra $\mathcal{A}$ on $\mathbb{N}$:

$$\mathcal{R} = \left\{ \begin{array}{c} \mathsf{a}(x, \mathsf{o}) \to x \\ \mathsf{a}(x, \mathsf{s}(y)) \to \mathsf{s}(\mathsf{a}(x, y)) \end{array} \right\} \qquad \mathcal{A}\colon \left\{ \begin{array}{r} \mathsf{o}_\mathcal{A} = 1 \\ \mathsf{s}_\mathcal{A}(x) = x + 1 \\ \mathsf{a}_\mathcal{A}(x, y) = x + 2y \end{array} \right.$$

termination of $\mathcal{R}$ is shown as follows:

$$\begin{aligned} \mathcal{R} \subseteq >_\mathcal{A} &\iff \bigwedge \left\{ \begin{array}{ll} \forall x \in \mathbb{N}. & 2 + x > 0 + x \\ \forall x, y \in \mathbb{N}. & 2 + x + 2y > 1 + x + 2y \end{array} \right\} \\ &\iff \bigwedge \left\{ \begin{array}{lll} 2 > 0, & 1 \geqslant 1 & \\ 2 > 1, & 1 \geqslant 1, & 2 \geqslant 2 \end{array} \right\} \\ &\iff \text{true} \end{aligned}$$

Finding Interpretations Automatically

consider TRS $\mathcal{R}$ and algebra $\mathcal{A}$ on $\mathbb{N}$

$$\mathcal{R} = \left\{ \begin{array}{c} \mathsf{a}(x, \mathsf{o}) \to x \\ \mathsf{a}(x, \mathsf{s}(y)) \to \mathsf{s}(\mathsf{a}(x, y)) \end{array} \right\} \qquad \mathcal{A}\colon \left\{ \begin{array}{r} \mathsf{o}_\mathcal{A} = o_0 \\ \mathsf{s}_\mathcal{A}(x) = s_0 + s_1 x \\ \mathsf{a}_\mathcal{A}(x, y) = a_0 + a_1 x + a_2 y \end{array} \right.$$

with $o_0, s_0, a_0 \in \mathbb{N}$ and $s_1, a_1, a_2 \in \mathbb{N}_+$ (why?):

$$\begin{aligned} &\mathcal{R} \subseteq >_\mathcal{A} \\ \iff &\bigwedge \left\{ \begin{array}{ll} \forall x \in \mathbb{N}. & a_0 + a_1 x + a_2 o_0 > x \\ \forall x, y \in \mathbb{N}. & a_0 + a_1 x + a_2 s_0 + a_2 s_1 y > s_0 + s_1 a_0 + s_1 a_1 x + s_1 a_2 y \end{array} \right\} \\ \iff &\bigwedge \left\{ \begin{array}{lll} a_0 + a_2 o_0 > 0, & a_1 \geqslant 1 & \\ a_0 + a_2 s_0 > s_0 + s_1 a_0, & a_1 \geqslant s_1 a_1, & a_2 s_1 \geqslant s_1 a_2 \end{array} \right\} \end{aligned}$$

Solving Constraints by SMT Solvers

```
(declare-const o0 Int)
(declare-const s0 Int)
(declare-const s1 Int)
(declare-const a0 Int)
(declare-const a1 Int)
(declare-const a2 Int)
```

```
(assert (and
  (>= o0 0)
  (>= s0 0) (>= s1 1)
  (>= a0 0) (>= a1 1) (>= a2 1)
  (> (+ a0 (* a2 o0)) 0)
  (>= a1 1)
  (> (+ a0 (* a2 s0)) (+ s0 (* s1 a0)))
  (>= a1 (* s1 a1))
  (>= (* a2 s1) (* s1 a2))))
(check-sat)
(get-value (o0 s0 s1 a0 a1 a2))
```

output:

```
sat
((o0 11) (s0 13) (s1 1) (a0 1) (a1 12) (a2 8))
```


Linear Interpretations

Definition

linear interpretations are of form

$$f_\mathcal{A}(x_1, \ldots, x_n) = f_0 + f_1 x_1 + \cdots + f_n x_n \quad (f_0, f_1, \ldots, f_n \in \mathbb{N})$$

Theorem

solving polynomial equations over $\mathbb{Z}$ is undecidable

Theorem (Mitterwallner et al. 2024)

termination by linear interpretations is undecidable

Contents: Proving (Non-)Termination Automatically

TRS category in termCOMP 2024:

tool/method	YES	NO
AProVE	1030	282
NaTT	876	169
MU-TERM	717	135
NTI	293	275
autonon	—	233
1 **linear interpretations**	145	—
2 **dependency pair method with linear and max/plus**	431	—
3 **termination of imperative programs**	—	—
4 **loop detection**	—	$\geqslant$ 178

Derivational Complexity

Definition (Hofbauer and Lautemann 1989)

- $\mathsf{dh}(t, \to) = \max\{n \in \mathbb{N} \mid t \to^n u \text{ for some } u\}$
- $\mathsf{dc}_{\mathcal{R}}(n) = \max\{\mathsf{dh}(t, \to_{\mathcal{R}}) \mid t \text{ is term with } |t| \leqslant n\}$

Theorem

$\mathsf{dc}_{\mathcal{R}}(n)$ is bound from above by exponetial function if
$\mathcal{R} \subseteq >_{\mathcal{A}}$ for some linear interpretation $\mathcal{A}$ on $\mathbb{N}$

Complexity Analysis of Bubble Sort Algorithm

TRS $\mathcal{R}$

$$\mathsf{b}(\mathsf{a}(x)) \to \mathsf{a}(\mathsf{b}(x))$$

take linear interpretation $\mathcal{A}$ on $\mathbb{N}$:

$$\mathsf{a}_{\mathcal{A}}(x) = x + 1 \qquad \mathsf{b}_{\mathcal{A}}(x) = 2x$$

thus, $\mathsf{dc}_{\mathcal{R}}(n)$ is boudned by exponetial function

but bubble sort is quadratic-time algorithm, isn't it?

Idea: Counting a below b

TRS $\mathcal{R}$

$$\mathsf{b}(\mathsf{a}(x)) \to \mathsf{a}(\mathsf{b}(x))$$

rewriting

$$\begin{array}{rll}
& \underset{2}{\underline{\mathsf{b}}}(\mathsf{a}(\underset{1}{\underline{\mathsf{b}}}(\mathsf{a}(\underset{0}{\underline{\mathsf{b}}}(x))))) & 2+1+0=3 \\
\to_{\mathcal{R}} & \mathsf{a}(\underset{1}{\underline{\mathsf{b}}}(\underset{1}{\underline{\mathsf{b}}}(\mathsf{a}(\underset{0}{\underline{\mathsf{b}}}(x))))) & 1+1+0=2 \\
\to_{\mathcal{R}} & \mathsf{a}(\underset{1}{\underline{\mathsf{b}}}(\mathsf{a}(\underset{0}{\underline{\mathsf{b}}}(\underset{0}{\underline{\mathsf{b}}}(x))))) & 1+0+0=1 \\
\to_{\mathcal{R}} & \mathsf{a}(\mathsf{a}(\underset{0}{\underline{\mathsf{b}}}(\underset{0}{\underline{\mathsf{b}}}(\underset{0}{\underline{\mathsf{b}}}(x))))) & 0+0+0=0
\end{array}$$

Complexity Analysis by Matrix Interpretations

Definition

triangular matrix interpretation $\mathcal{A}$ on $\mathbb{N}^2$ is of form

$$f_{\mathcal{A}}(\mathbf{x_1}, \ldots, \mathbf{x_n}) = \mathbf{f}_0 + F_1\mathbf{x_1} + \cdots + F_n\mathbf{x_n} \quad \text{with } \mathbf{f}_0 \in \mathbb{N}^2 \text{ and } F_i = \begin{pmatrix} 1 & a \\ 0 & b \end{pmatrix}$$

$(x, y) > (x', y')$ if $x > x'$ and $y \geqslant y'$

Theorem (Moser, Schnabl, and Waldmann 2010)

$\mathsf{dc}_{\mathcal{R}}(n) \in O(n^2)$ *if* $\mathcal{R} \subseteq >_{\mathcal{A}}$ *for some triangular matrix interpretation* $\mathcal{A}$ *on* $\mathbb{N}^2$

Exercise: Find Triangular Matrix Interpretation

TRS $\mathcal{R}$

$$\mathsf{b}(\mathsf{a}(x)) \to \mathsf{a}(\mathsf{b}(x))$$

take triangular matrix interpretation $\mathcal{A}$ on $\mathbb{N}^2$:

$$\mathsf{a}_{\mathcal{A}}(\mathbf{x}) = \mathbf{x} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} \qquad \mathsf{b}_{\mathcal{A}}(\mathbf{x}) = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \mathbf{x}$$

thus, $\mathsf{dc}_{\mathcal{R}}(n) \in O(n^2)$

Dependency Pair Method

Literature

- Arts and Giesl
  Termination of Term Rewriting Using Dependency Pairs
  TCS 236:133–178, 2000
- Hirokawa and Middeldorp
  Automating the Dependency Pair Method
  I&C 199(1,2), pp. 172–199, 2005
- Giesl, Thiemann, and Schneider-Kamp
  The Dependency Pair Framework: Combining Techniques for Automated Termination Proofs, LPAR-11, LNAI 3452, 301–331, 2005

Dependency Pair Method

Definition

- $t^\sharp = f^\sharp(t_1, \ldots, t_n)$ if $t = f(t_1, \ldots, t_n)$, where $f^\sharp$ is fresh symbol
- $\mathsf{DP}(\mathcal{R}) = \{\ell^\sharp \to t^\sharp \mid \ell \to r \in \mathcal{R},\ r \trianglerighteq t,\ \mathsf{root}(t) \in \mathcal{D}_\mathcal{R},\ \text{and}\ \ell \ntrianglerighteq t\}$

Example

$$\mathcal{R} = \left\{\begin{array}{c} \mathsf{ack}(0, y) \to y \\ \mathsf{ack}(\mathsf{s}(x), 0) \to \mathsf{ack}(x, \mathsf{s}(0)) \\ \mathsf{ack}(\mathsf{s}(x), \mathsf{s}(y)) \to \mathsf{ack}(x, \mathsf{ack}(\mathsf{s}(x), y)) \end{array}\right\}$$

$$\mathsf{DP}(\mathcal{R}) = \left\{\begin{array}{l} \mathsf{ack}^\sharp(\mathsf{s}(x), 0) \to \mathsf{ack}^\sharp(x, \mathsf{s}(0)) \\ \mathsf{ack}^\sharp(\mathsf{s}(x), \mathsf{s}(y)) \to \mathsf{ack}^\sharp(x, \mathsf{ack}(\mathsf{s}(x), y)) \\ \mathsf{ack}^\sharp(\mathsf{s}(x), \mathsf{s}(y)) \to \mathsf{ack}^\sharp(\mathsf{s}(x), y) \end{array}\right\}$$

Finiteness

Definition

$(\mathcal{P}, \mathcal{R})$ is finite if it has no infinite sequence

$$s_1 \xrightarrow{\epsilon}_\mathcal{P} t_1 \to^*_\mathcal{R} s_2 \xrightarrow{\epsilon}_\mathcal{P} t_2 \to^*_\mathcal{R} \cdots$$

that each t_i is $\mathcal{R}$-terminating

Theorem (Giesl et al. 2004)

$\mathcal{R}$ is terminating if and only if $(\mathsf{DP}(\mathcal{R}), \mathcal{R})$ is finite

Fact

$(\mathcal{P}, \mathcal{R})$ is finite if $\mathcal{P} = \varnothing$

how to prove finiteness when $\mathcal{P} \neq \varnothing$?

Subterm Criterion

Definition

- simple projection π maps $f^\sharp$ to one of argument positions
- $\pi(f^\sharp(t_1, \ldots, t_n)) = t_i$ if $\pi(f^\sharp) = i$
- $s \trianglerighteq^\pi t$ if $\pi(s) \trianglerighteq \pi(t)$
- $s \triangleright^\pi t$ if $\pi(s) \triangleright \pi(t)$

Example

if $\pi(\mathsf{ack}^\sharp) = 1$ then $\mathsf{ack}^\sharp(\mathsf{s}(x), \mathsf{s}(y)) \triangleright^\pi \mathsf{ack}^\sharp(x, \mathsf{ack}(\mathsf{s}(x), y))$

Theorem (Hirokawa and Middeldorp 2007)

if simple projection π satisfies $\mathcal{P} \subseteq \trianglerighteq^\pi$ then

$$(\mathcal{P}, \mathcal{R}) \text{ is finite} \iff (\mathcal{P} \setminus \triangleright^\pi, \mathcal{R}) \text{ is finite}$$

consider TRS $\mathcal{R}$

$$\begin{aligned} \mathsf{ack}(0, y) &\to y \\ \mathsf{ack}(\mathsf{s}(x), 0) &\to \mathsf{ack}(x, \mathsf{s}(0)) \\ \mathsf{ack}(\mathsf{s}(x), \mathsf{s}(y)) &\to \mathsf{ack}(x, \mathsf{ack}(\mathsf{s}(x), y)) \end{aligned}$$

$\mathsf{DP}(\mathcal{R})$ consists of

$$\begin{aligned} \mathsf{ack}^\sharp(\mathsf{s}(x), 0) &\xrightarrow{\mathbf{1}} \mathsf{ack}^\sharp(x, \mathsf{s}(0)) \\ \mathsf{ack}^\sharp(\mathsf{s}(x), \mathsf{s}(y)) &\xrightarrow{\mathbf{2}} \mathsf{ack}^\sharp(x, \mathsf{ack}(\mathsf{s}(x), y)) \\ \mathsf{ack}^\sharp(\mathsf{s}(x), \mathsf{s}(y)) &\xrightarrow{\mathbf{3}} \mathsf{ack}^\sharp(\mathsf{s}(x), y) \end{aligned}$$

$\boxed{1}$ subterm criterion with $\pi(\mathsf{ack}^\sharp) = 1$ removes $\mathbf{1}$ and $\mathbf{2}$
$\boxed{2}$ subterm criterion with $\pi(\mathsf{ack}^\sharp) = 2$ removes $\mathbf{3}$
$\boxed{3}$ no dependency pair remains; hence $\mathcal{R}$ is terminating

Limitation of Subterm Criterion

we show termination of TRS $\mathcal{R}$

$$\begin{aligned} x - 0 &\to x \\ \mathsf{s}(x) - \mathsf{s}(y) &\to x - y \\ 0 \div \mathsf{s}(y) &\to 0 \\ \mathsf{s}(x) \div \mathsf{s}(y) &\to \mathsf{s}((x - y) \div \mathsf{s}(y)) \end{aligned}$$

DP($\mathcal{R}$) consists of

$$\begin{aligned} \mathsf{s}(x) -^\sharp \mathsf{s}(y) &\to x -^\sharp y \\ \mathsf{s}(x) \div^\sharp \mathsf{s}(y) &\to (x - y) \div^\sharp \mathsf{s}(y) \\ \mathsf{s}(x) \div^\sharp \mathsf{s}(y) &\to x -^\sharp y \end{aligned}$$

semantically decreasing! ☞ revisit interpretation method

Reduction Pairs and Removal of Dependency Pairs

Definition

$(\gtrsim, >)$ is reduction pair if

- $\gtrsim$ is preorder that is closed under contexts and substitutions
- $>$ is well-founded order that is closed under substitutions, and
- $\gtrsim \cdot > \cdot \gtrsim \subseteq >$

Fact

$(\geqslant_\mathcal{A}, >_\mathcal{A})$ is reduction pair if $\mathcal{A}$ is well-founded weakly monotone algebras

Theorem

if $(\gtrsim, >)$ is reduction pair and $\mathcal{R} \subseteq \gtrsim$ then

$$(\mathcal{P}, \mathcal{R}) \text{ is finite} \iff (\mathcal{P} \setminus >, \mathcal{R}) \text{ is finite}$$

we show termination of TRS $\mathcal{R}$

$x - 0 \to x$	$x \geqslant x$
$\mathsf{s}(x) - \mathsf{s}(y) \to x - y$	$x + 1 \geqslant x + 1$
$0 \div \mathsf{s}(y) \to 0$	$0 \geqslant 0$
$\mathsf{s}(x) \div \mathsf{s}(y) \to \mathsf{s}((x - y) \div \mathsf{s}(y))$	$x + 1 \geqslant x + 1$

DP($\mathcal{R}$) consists of

$\mathsf{s}(x) -^\sharp \mathsf{s}(y) \xrightarrow{\mathbf{1}} x -^\sharp y$	$x + 1 > x$
$\mathsf{s}(x) \div^\sharp \mathsf{s}(y) \xrightarrow{\mathbf{2}} (x - y) \div^\sharp \mathsf{s}(x)$	$x + 1 > x$
$\mathsf{s}(x) \div^\sharp \mathsf{s}(y) \xrightarrow{\mathbf{3}} x -^\sharp y$	$x + 1 > x$

1. well-founded weakly monotone algebra $\mathcal{A}$ on $\mathbb{N}$ removes $\mathbf{1}$, $\mathbf{2}$, and $\mathbf{3}$
   $0_\mathcal{A} = 0 \quad \mathsf{s}_\mathcal{A}(x) = x + 1 \quad x -_\mathcal{A} y = x -^\sharp_\mathcal{A} y = x \div_\mathcal{A} y = x \div^\sharp_\mathcal{A} y = x$
2. no dependency pairs remain, and hence $\mathcal{R}$ is terminating

Exercise: Prove Termination!

TRS $\mathcal{R}$

$x + 0 \to x$	$x \geqslant x$
$x + \mathsf{s}(y) \to \mathsf{s}(x + y)$	$\max\{x, y\} \geqslant \max\{x, y\}$
$x \times 0 \to 0$	$x \geqslant 0$
$x \times \mathsf{s}(y) \to (x \times y) + x$	$\max\{x, y\} \geqslant \max\{x, y\}$

DP($\mathcal{R}$) consists of

$x +^\sharp \mathsf{s}(y) \to x +^\sharp y$	$0 \geqslant 0$	$\mathsf{s}(y) \rhd y$
$x \times^\sharp \mathsf{s}(y) \to (x \times y) +^\sharp x$	$1 > 0$	
$x \times^\sharp \mathsf{s}(y) \to x \times^\sharp y$	$1 \geqslant 1$	$\mathsf{s}(y) \rhd y$

Recommendable Interpretations

- $0, 1$-coefficient linear interpretation $\mathcal{A}$ on $\mathbb{N}$:

$$f_{\mathcal{A}}(x_1, \dots, x_n) = f_0 + f_1 x_1 + \cdots + f_n x_n$$

with $f_0 \in \mathbb{N}$ and $f_1, \dots, f_n \in \{0, 1\}$

- max/plus interpretation $\mathcal{A}$ on $\mathbb{N}$:

$$f_{\mathcal{A}}(x_1, \dots, x_n) = \max\{f_0, f'_1(f_1 + x_1), \dots, f'_n(f_n + x_n)\}$$

with $f_0 \in \mathbb{N}$, $f_1, \dots, f_n \in \mathbb{Z}$, and $f'_1, \dots, f'_n \in \{0, 1\}$

Contents: Proving (Non-)Termination Automatically

TRS category in termCOMP 2024:

	tool/method	YES	NO
	AProVE	1030	282
	NaTT	876	169
	MU-TERM	717	135
	NTI	293	275
	autonon	—	233
1	**linear interpretations**	145	—
2	**dependency pair method with linear and max/plus**	431	—
3	**termination of imperative programs**	—	—
4	**loop detection**	—	$\geqslant 178$

Termination of Imperative Programs

From C to Conditional TRS (CTRS)

see e.g. Fuhs, Kop, and Nishida 2017

	C program	CTRS
	`int` sum(x)	
	{	
sum:	`int` $y = 0$;	$\mathsf{sum}(x) \to \mathsf{f}_1(x, 0)$
f_1:	`while` $(x > 0)$	$\mathsf{f}_1(x, y) \to \mathsf{f}_2(x, y)$ if $x > 0 \to^* \mathsf{true}$
	{	$\mathsf{f}_1(x, y) \to \mathsf{f}_4(x, y)$ if $x > 0 \to^* \mathsf{false}$
f_2:	$y = x + y$;	$\mathsf{f}_2(x, y) \to \mathsf{f}_3(x, x + y)$
f_3:	$x = x - 1$;	$\mathsf{f}_3(x, y) \to \mathsf{f}_1(x - \mathsf{s}(0), y)$
	}	
f_4:	`return` y;	$\mathsf{f}_4(x, y) \to y$
	}	

Unraveling CTRS into TRS

Example

$$\begin{array}{rll}
x + 0 \to x & \mathsf{sum}(x) \to \mathsf{f}_1(x, 0) & \\
x + \mathsf{s}(y) \to \mathsf{s}(x + y) & \mathsf{f}_1(x, y) \to \mathsf{g}_1(x > 0, x, y) & \\
x - 0 \to x & \mathsf{g}_1(\mathsf{true}, x, y) \to \mathsf{f}_2(x, y) & \text{if } x > 0 \to^* \mathsf{true} \\
\mathsf{s}(x) - \mathsf{s}(y) \to x - y & \mathsf{g}_1(\mathsf{false}, x, y) \to \mathsf{f}_4(x, y) & \text{if } x > 0 \to^* \mathsf{false} \\
0 > x \to \mathsf{false} & \mathsf{f}_2(x, y) \to \mathsf{f}_3(x, x + y) & \\
\mathsf{s}(x) > 0 \to \mathsf{true} & \mathsf{f}_3(x, y) \to \mathsf{f}_1(x - \mathsf{s}(0), y) & \\
\mathsf{s}(x) > \mathsf{s}(y) \to x > y & \mathsf{f}_4(x, y) \to y &
\end{array}$$

Fact (Ohlebusch 2001)

deterministic 3-CTRS is operationally terminating if unraveled TRS is terminating

Can Termination Tools Prove Termination?

$$\begin{array}{rr}
x + 0 \to x & \mathsf{sum}(x) \to \mathsf{f}_1(x, 0) \\
x + \mathsf{s}(y) \to \mathsf{s}(x + y) & \mathsf{f}_1(x, y) \to \mathsf{g}_1(x > 0, x, y) \\
x - 0 \to x & \mathsf{g}_1(\mathsf{true}, x, y) \to \mathsf{f}_2(x, y) \\
\mathsf{s}(x) - \mathsf{s}(y) \to x - y & \mathsf{g}_1(\mathsf{false}, x, y) \to \mathsf{f}_4(x, y) \\
0 > x \to \mathsf{false} & \mathsf{f}_2(x, y) \to \mathsf{f}_3(x, x + y) \\
\mathsf{s}(x) > 0 \to \mathsf{true} & \mathsf{f}_3(x, y) \to \mathsf{f}_1(x - \mathsf{s}(0), y) \\
\mathsf{s}(x) > \mathsf{s}(y) \to x > y & \mathsf{f}_4(x, y) \to y
\end{array}$$

AProVE	NaTT	TTT2
YES	YES	MAYBE

Contents: Proving (Non-)Termination Automatically

TRS category in termCOMP 2024:

tool/method	YES	NO
AProVE	1030	282
NaTT	876	169
MU-TERM	717	135
NTI	293	275
autonon	—	233
1 **linear interpretations**	145	—
2 **dependency pair method with linear and max/plus**	431	—
3 **termination of imperative programs**	—	—
4 **loop detection**	—	$\geqslant$ 178

Non-Termination

Looping

Fact

TRS $\mathcal{R}$ is non-terminating if it admits loop $t \to_{\mathcal{R}}^{+} C[t\sigma]$

Example

1. $\mathcal{R} = \{\ \mathsf{from}(x) \to x : \mathsf{from}(\mathsf{s}(x))\ \}$
2. $\mathcal{R} = \{\ \mathsf{f}(x, \mathsf{s}(y)) \to \mathsf{f}(\mathsf{s}(x), x)\ \}$
3. $\mathcal{R} = \{\ \mathsf{ab} \to \mathsf{bbaa}\ \}$ $\qquad \mathsf{abb} \to_{\mathcal{R}} \mathsf{bbaab} \to_{\mathcal{R}} \mathsf{bbabbaa}$
4. $\mathcal{R} = \left\{ \begin{array}{l} \mathsf{f}(\mathsf{a}, \mathsf{b}, x) \to \mathsf{f}(x, x, x) \\ \quad \mathsf{g}(x, y) \to x \\ \quad \mathsf{g}(x, y) \to y \end{array} \right\}$ $\qquad \mathsf{f}(\mathsf{a}, \mathsf{b}, \mathsf{g}(\mathsf{a}, \mathsf{b})) \to_{\mathcal{R}}^{3} \mathsf{f}(\mathsf{a}, \mathsf{b}, \mathsf{g}(\mathsf{a}, \mathsf{b}))$

Example beyond Self-Reducible Rule

TRS $\mathcal{R}$

$$\mathsf{f}(x, \mathsf{s}(y)) \to \mathsf{f}(\mathsf{s}(x), x)$$

is non-terminating because it admits loop:

$$\mathsf{f}(\mathsf{s}(z), \mathsf{s}(y)) \to_{\mathcal{R}} \mathsf{f}(\mathsf{s}(\underbrace{\mathsf{s}(z)}_{z}), \mathsf{s}(\underbrace{z}_{y}))$$

how to find such loop? ☞ semi-unification

Loop Detection by Semi-Unifiability

Definition

(s, t) is semi-unifiable if $s\sigma\tau = t\sigma$ for some σ and τ

Kapur et al. 1991

TRS $\mathcal{R}$ is non-terminating if $s \to_{\mathcal{R}}^{+} C[t]$ for some semi-unifiable pair (s, t)

Proof.

if $s \to_{\mathcal{R}}^{+} C[t]$ and $s\sigma\tau = t\sigma$ then $s\sigma \to_{\mathcal{R}}^{+} C\sigma[t\sigma] = C\sigma[s\sigma\tau]$ $\square$

Note

semi-unification algorithm exists see Aoto and Iwami 2013

Example for Semi-Unification

rule $\mathsf{f}(x, \mathsf{s}(y)) \to \mathsf{f}(\mathsf{s}(x), x)$ is semi-unifiable:

$$\begin{array}{rrcl} & \mathsf{f}(x\sigma\tau, \mathsf{s}(y\sigma\tau)) = \mathsf{f}(\mathsf{s}(x\sigma), x\sigma) & & \\ \iff & x\sigma\tau = \mathsf{s}(x\sigma) & \wedge & \mathsf{s}(y\sigma\tau) = x\sigma \\ \iff & \mathsf{s}(y\sigma\tau\tau) = \mathsf{s}(\mathsf{s}(y\sigma\tau)) & \wedge & x\sigma = \mathsf{s}(y\sigma\tau) \\ \iff & \underbrace{\underbrace{\underbrace{y\sigma}_{y}\tau}_{z}\tau}_{\mathsf{s}(z)} = \underbrace{\mathsf{s}(\underbrace{\underbrace{y\sigma}_{y}\tau}_{z})}_{\mathsf{s}(z)} & \wedge & \underbrace{x\sigma}_{\mathsf{s}(z)} = \underbrace{\mathsf{s}(\underbrace{\underbrace{y\sigma}_{y}\tau}_{z})}_{\mathsf{s}(z)} \end{array}$$

therefore, $\sigma = \{x \mapsto \mathsf{s}(z)\}$ and $\tau = \{y \mapsto z,\ y \mapsto \mathsf{s}(z)\}$, which yield loop:

$$\mathsf{f}(\mathsf{s}(z), \mathsf{s}(y)) \to_{\mathcal{R}} \mathsf{f}(\mathsf{s}(\mathsf{s}(z)), \mathsf{s}(z))$$

Loop Consisting of Many-Steps

one-rule TRS $\mathcal{R}$

$$\mathsf{a}(\mathsf{b}(x)) \to \mathsf{b}(\mathsf{b}(\mathsf{a}(\mathsf{a}(x))))$$

admits loop:

$$\mathsf{a}(\mathsf{b}(\mathsf{b}(x))) \to_{\mathcal{R}} \mathsf{b}(\mathsf{b}(\mathsf{a}(\mathsf{a}(\mathsf{b}(x))))) \to_{\mathcal{R}} \mathsf{b}(\mathsf{b}(\mathsf{a}(\mathsf{b}(\mathsf{b}(\mathsf{a}(\mathsf{a}(x)))))))$$

how to find such loop? ☞ narrowing

Forward Narrowing

Definition

$\ell_1\sigma \to r_1[r_2]\sigma$ is narrowed rule of $\mathcal{R}$ if

- $\ell_1 \to r_1$ and $\ell_2 \to r_2$ are variants of $\mathcal{R}$-rules
- $p \in \mathcal{P}\mathsf{os}(r_1)$, and $\sigma = \mathrm{mgu}(r_1|_p, \ell_2)$

Theorem

$\mathcal{R}$ and $\mathcal{R} \cup \{\ell \to r\}$ are equi-terminating if $\ell \to r$ is narrowed rule of $\mathcal{R}$

Proof.

immediate from $\ell \to_{\mathcal{R}} r$ or $\ell \to^2_{\mathcal{R}} r$ □

Example for Narrowing

consider TRS

$$\mathsf{ab}x \to \mathsf{bba}\underbrace{\mathsf{a}x}_{\mathsf{ab}x \to \mathsf{bbaa}x}$$

narrowing yields new rule:

$$\mathsf{abb}x \to \mathsf{bbabbaa}x$$

since it admits loop, TRS is non-terminating

Non-termination Proof of Toyama's Example

TRS $\mathcal{R}$

$$\mathsf{f}(\mathsf{a}, \mathsf{b}, x) \to \mathsf{f}(\underbrace{x}_{\mathsf{g}(x,y)\to x}, x, x) \qquad \mathsf{g}(x, y) \to x \qquad \mathsf{g}(x, y) \to y$$

narrowing yields rule:

$\boxed{1}$ $\mathsf{f}(\mathsf{a}, \mathsf{b}, \mathsf{g}(x, y)) \to \mathsf{f}(x, \underbrace{\mathsf{g}(x, y)}_{\mathsf{g}(x,y)\to y}, \mathsf{g}(x, y))$

$\boxed{2}$ $\underbrace{\mathsf{f}(\mathsf{a}, \mathsf{b}, \mathsf{g}(x, y)) \to \mathsf{f}(x, y, \mathsf{g}(x, y))}_{\text{semi-unifiable}}$

thus, $\mathcal{R}$ is non-terminating due to loop $\mathsf{f}(\mathsf{a}, \mathsf{b}, \mathsf{g}(\mathsf{a}, \mathsf{b})) \to^+_{\mathcal{R}} \mathsf{f}(\mathsf{a}, \mathsf{b}, \mathsf{g}(\mathsf{a}, \mathsf{b}))$

Automation of Non-Termination Analysis

for instance,

1. perform narrowing twice
2. rewrite right-hand sides of rules (e.g.) 5 times
3. check if rule admits loop by using semi-unification

Contents: Proving (Non-)Termination Automatically

TRS category in termCOMP 2024:

	tool/method	YES	NO
	AProVE	1030	282
	NaTT	876	169
	MU-TERM	717	135
	NTI	293	275
	autonon	—	233
1	**linear interpretations**	145	—
2	**dependency pair method with linear and max/plus**	431	—
3	**termination of imperative programs**	—	—
4	**loop detection**	—	$\geqslant$ 178

Summary

- linear interpretations
- dependency pairs
- termination of imperative programs (still challenging)
- loop detection

thanks for your attention!