Approximations for Strategies and Termination

Aart Middeldorp

University of Tsukuba

OVERVIEW

- term rewriting
- call by need computations
- approximations for strategies

} Durand & Middeldorp CADE'97

- dependency pairs
- approximations for termination

} Middeldorp IJCAR'01

TERM REWRITING

signature 0 constant s unary $+$ $\times$ binary

rewrite rules

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

TRS

rewriting

$$\begin{aligned} \mathsf{s}(\underline{\mathsf{s}(0) \times \mathsf{s}(\mathsf{s}(0))}) &\to \mathsf{s}(\underline{0 \times \mathsf{s}(\mathsf{s}(0))} + \mathsf{s}(\mathsf{s}(0))) \\ &\to \mathsf{s}(\underline{0 + \mathsf{s}(\mathsf{s}(0))}) \\ &\to \mathsf{s}(\mathsf{s}(\mathsf{s}(0))) \end{aligned}$$

redex

normal form

signature 0 fib constants s unary $+$ nth f $:$ binary

rewrite rules

$$\begin{aligned} 0 + y &\to y & \mathsf{fib} &\to \mathsf{f}(\mathsf{s}(0), \mathsf{s}(0)) \\ \mathsf{s}(x) + y &\to \mathsf{s}(x + y) & \mathsf{f}(x, y) &\to x : \mathsf{f}(y, x + y) \\ \mathsf{nth}(0, y : z) &\to y & \mathsf{nth}(\mathsf{s}(x), y : z) &\to \mathsf{nth}(x, z) \end{aligned}$$

rewriting

$$\begin{aligned} \mathsf{nth}(\mathsf{s}(0), \underline{\mathsf{fib}}) &\to \mathsf{nth}(\mathsf{s}(0), \underline{\mathsf{f}(\mathsf{s}(0), \mathsf{s}(0))}) \\ &\to \underline{\mathsf{nth}(\mathsf{s}(0), \mathsf{s}(0) : \mathsf{f}(\mathsf{s}(0), \mathsf{s}(0) + \mathsf{s}(0)))} \\ &\to \mathsf{nth}(0, \mathsf{f}(\mathsf{s}(0), \underline{\mathsf{s}(0) + \mathsf{s}(0)})) \\ &\to \mathsf{nth}(0, \mathsf{f}(\mathsf{s}(0), \mathsf{s}(\underline{0 + \mathsf{s}(0)}))) \\ &\to \mathsf{nth}(0, \underline{\mathsf{f}(\mathsf{s}(0), \mathsf{s}(\mathsf{s}(0)))}) \\ &\to \underline{\mathsf{nth}(0, \mathsf{s}(0) : \mathsf{f}(\mathsf{s}(\mathsf{s}(0)), \mathsf{s}(0) + \mathsf{s}(\mathsf{s}(0))))} \\ &\to \mathsf{s}(0) \end{aligned}$$

$$\mathsf{nth}(\mathsf{s}(0), \underline{\mathsf{fib}}) \to^{\omega} \mathsf{nth}(\mathsf{s}(0), \mathsf{s}(0) : (\mathsf{s}(0) : (\mathsf{s}^2(0) : (\mathsf{s}^3(0) : (\mathsf{s}^5(0) : \cdots)))))$$

STRATEGIES

DEFINITION

- ➜ strategy selects redexes
  - ➜ leftmost outermost
  - ➜ parallel outermost
- ➜ strategy is normalizing if it computes normal forms for all terms that admit normal forms

THEOREM O'Donnell LNCS'77

for orthogonal TRSs parallel outermost strategy is normalizing

OBSERVATION

parallel outermost is not optimal because it performs useless steps

EXAMPLE

TRS

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

$$\underline{(0 \times \mathsf{s}(0))} \times \underline{(0 + \mathsf{s}(0))} \xrightarrow{*} \underline{0 \times \mathsf{s}(0)} \to 0$$

$$\underline{(0 \times \mathsf{s}(0))} \times (0 + \mathsf{s}(0)) \to \underline{0 \times (0 + \mathsf{s}(0))} \to 0$$

redex $0 + \mathsf{s}(0)$ is not needed

NEEDEDNESS

DEFINITION

redex Δ in term t is needed if descendant of Δ is contracted in every rewrite sequence from t to normal form

NOTATION

$\mathcal{R}_\bullet = \mathcal{R} \cup \{\bullet \to \bullet\}$

EASY LEMMA

for orthogonal TRSs $\mathcal{R}$

redex Δ in $C[\Delta]$ is needed $\iff$ $\neg\, \exists\, t \in \mathsf{NF}(\mathcal{R}_\bullet)\colon\ C[\bullet] \xrightarrow[\mathcal{R}]{*} t$ (dot-free normal form)

COROLLARY

redex Δ in $C[\Delta]$ is needed $\iff$ redex Δ' in $C[\Delta']$ is needed

EXAMPLE

TRS

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

$$\underline{(0 \times \mathsf{s}(0))} \times \underline{(0 + \mathsf{s}(0))} \xrightarrow{*} \underline{0 \times \mathsf{s}(0)} \to 0$$

$$\underline{(0 \times \mathsf{s}(0))} \times (0 + \mathsf{s}(0)) \to \underline{0 \times (0 + \mathsf{s}(0))} \to 0$$

redex $0 + \mathsf{s}(0)$ is not needed

$\underline{(0 \times \mathsf{s}(0))} \times \bullet \to \underline{0 \times \bullet} \to 0$ (dot-free normal form)

THEOREM Huet & Lévy '79 '91

for orthogonal TRSs

- ➜ every reducible term has needed redex
- ➜ needed reduction is normalizing

PROBLEM

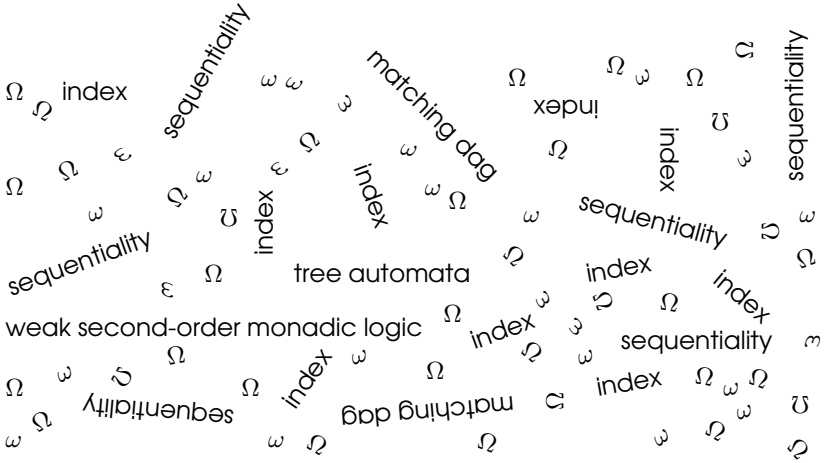
for orthogonal TRSs it is undecidable whether redex is needed

easy reduction from halting problem for Turing machines

DECIDABILITY

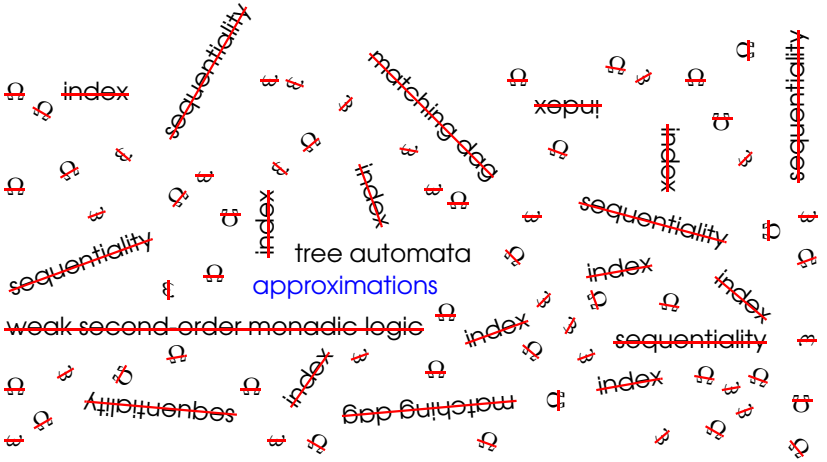
Huet & Lévy '79 '91

DECIDABILITY



Huet & Lévy '79 '91 Comon LICS'95

DECIDABILITY



Huet & Lévy '79 '91 Comon LICS'95

Durand & Middeldorp CADE'97

NOTATION

$(\xrightarrow[\mathcal{R}]{*})[L]$ set of ground terms that rewrite in $\mathcal{R}$ to term in L

EASY LEMMA

for orthogonal TRSs $\mathcal{R}$

redex Δ in $C[\Delta]$ is needed $\iff \neg\,\exists\, t \in \mathrm{NF}(\mathcal{R}_\bullet)\colon\ C[\bullet] \xrightarrow[\mathcal{R}]{*} t$

$\iff C[\bullet] \notin \underbrace{(\xrightarrow[\mathcal{R}]{*})[\mathrm{NF}(\mathcal{R}_\bullet)]}_{\text{not computable}}$

REMARK

- ➜ $\mathrm{NF}(\mathcal{R}_\bullet)$ is regular tree language
- ➜ $(\xrightarrow[\mathcal{R}]{*})[\mathrm{NF}(\mathcal{R}_\bullet)]$ is not regular tree language

KEY IDEA

approximate $\mathcal{R}$ by TRS $\mathcal{S}$ such that

instance: term t
question: $t \in (\xrightarrow[\mathcal{S}]{*})[\mathrm{NF}(\mathcal{R}_\bullet)]$?

is decidable

DEFINITION

- ➜ TRS $\mathcal{S}$ approximates $\mathcal{R}$ if
  - ① $\mathcal{S}$ is left-linear
  - ② $\xrightarrow[\mathcal{R}]{*} \subseteq \xrightarrow[\mathcal{S}]{*}$
  - ③ $\mathrm{NF}(\mathcal{R}) = \mathrm{NF}(\mathcal{S})$
- ➜ redex Δ in $C[\Delta]$ is $\mathcal{S}$-needed if $C[\bullet] \in (\xrightarrow[\mathcal{S}]{*})[\mathrm{NF}(\mathcal{S}_\bullet)]$

LEMMA

if $\mathcal{S}$ approximates orthogonal TRS $\mathcal{R}$ then $\mathcal{S}$-needed redexes are needed

OBSERVATION

orthogonal TRS $\mathcal{R}$ admits computable call-by-need strategy
if it admits approximation $\mathcal{S}$ such that

- ① $\mathcal{S}$-needed redexes are computable
- ② every reducible term has $\mathcal{S}$-needed redex

APPROXIMATIONS

➜ Huet & Lévy	'79 '91	strong
➜ Oyamaguchi	SIAMJC'93	nv
➜ Comon	LICS'95	shallow
➜ Jacquemard	RTA'96	right-linear growing
➜ Nagaya & Toyama	RTA'99	growing
➜ Takai & Kaji & Seki	RTA'00	inverse finite path overlapping

Strong Approximation

Huet & Lévy '79 '91

$$\mathcal{R}\quad\begin{array}{rcl} 0+y & \to & y \\ \mathsf{s}(x)+y & \to & \mathsf{s}(x+y) \\ 0\times y & \to & 0 \\ \mathsf{s}(x)\times y & \to & x\times y+y \end{array}$$

$$\mathcal{R}_{\mathsf{s}}\quad\begin{array}{rcl} 0+y & \to & z \\ \mathsf{s}(x)+y & \to & z \\ 0\times y & \to & z \\ \mathsf{s}(x)\times y & \to & z \end{array}$$

linearize left-hand sides and
replace right-hand sides by fresh variables

Example

$$\mathcal{R}\quad\begin{array}{rcl} 0+y & \to & y \\ \mathsf{s}(x)+y & \to & \mathsf{s}(x+y) \end{array}\qquad\begin{array}{rcl} 0+y & \to & z \\ \mathsf{s}(x)+y & \to & z \end{array}\quad\mathcal{R}_{\mathsf{s}}$$

term $\underbrace{(0+\mathsf{s}(\Delta_1))}_{\Delta_3}+\Delta_2$

redexes	Δ_1	Δ_2	Δ_3
needed	◯	◯	◯
s	×	×	◯

$\mathcal{R}$ is non-erasing

$$\underline{(0+\mathsf{s}(\bullet))}+\Delta_2 \underset{\mathsf{s}}{\to} \underline{0+\Delta_2} \underset{\mathsf{s}}{\to} 0$$

$$\underline{(0+\mathsf{s}(\Delta_1))}+\bullet \underset{\mathsf{s}}{\to} \underline{0+\bullet} \underset{\mathsf{s}}{\to} 0$$

$$\bullet+\underline{\Delta_2} \underset{\mathsf{s}}{\overset{*}{\to}} \bullet+?$$

NV Approximation

Oyamaguchi SIAMJC'93

$$\mathcal{R}\quad\begin{array}{rcl} 0+y & \to & y \\ \mathsf{s}(x)+y & \to & \mathsf{s}(x+y) \\ 0\times y & \to & 0 \\ \mathsf{s}(x)\times y & \to & x\times y+y \end{array}$$

$$\mathcal{R}_{\mathsf{nv}}\quad\begin{array}{rcl} 0+y & \to & z \\ \mathsf{s}(x)+y & \to & \mathsf{s}(z+z') \\ 0\times y & \to & 0 \\ \mathsf{s}(x)\times y & \to & z\times z'+z'' \end{array}$$

linearize left-hand sides and
replace variables in right-hand sides by fresh variables

Example

$$\mathcal{R}\quad\begin{array}{rcl} 0+y & \to & y \\ \mathsf{s}(x)+y & \to & \mathsf{s}(x+y) \end{array}\qquad\begin{array}{rcl} 0+y & \to & z \\ \mathsf{s}(x)+y & \to & \mathsf{s}(z+z') \end{array}\quad\mathcal{R}_{\mathsf{nv}}$$

term $\underbrace{(0+\mathsf{s}(\Delta_1))}_{\Delta_3}+\Delta_2$

redexes	Δ_1	Δ_2	Δ_3
needed	◯	◯	◯
s	×	×	◯
nv	×	×	◯

$$\underline{(0+\mathsf{s}(\bullet))}+\Delta_2 \underset{\mathsf{nv}}{\to} \underline{0+\Delta_2} \underset{\mathsf{nv}}{\to} 0$$

$$\underline{(0+\mathsf{s}(\Delta_1))}+\bullet \underset{\mathsf{nv}}{\to} \underline{0+\bullet} \underset{\mathsf{nv}}{\to} 0$$

$$\bullet+\underline{\Delta_2} \underset{\mathsf{nv}}{\overset{*}{\to}} \bullet+?$$

GROWING APPROXIMATION

Jacquemard RTA'96
Nagaya & Toyama RTA'99

$$\mathcal{R}: \quad \begin{array}{rcl} 0+y & \to & y \\ \mathsf{s}(x)+y & \to & \mathsf{s}(x+y) \\ 0\times y & \to & 0 \\ \mathsf{s}(x)\times y & \to & x\times y+y \end{array}$$

$$\mathcal{R}_{\mathsf{g}}: \quad \begin{array}{rcl} 0+y & \to & y \\ \mathsf{s}(x)+y & \to & \mathsf{s}(z+y) \\ 0\times y & \to & 0 \\ \mathsf{s}(x)\times y & \to & z\times y+y \end{array}$$

linearize left-hand sides and rename variables in right-hand sides that occur at depth > 1 in left-hand sides by fresh variables

EXAMPLE

$\mathcal{R}$: $0+y \to y$, $\mathsf{s}(x)+y \to \mathsf{s}(x+y)$ $\qquad$ $\mathcal{R}_{\mathsf{g}}$: $0+y \to y$, $\mathsf{s}(x)+y \to \mathsf{s}(z+y)$

term $\underbrace{(0+\mathsf{s}(\Delta_1))}_{\Delta_3}+\Delta_2$

redexes	Δ_1	Δ_2	Δ_3
needed	○	○	○
s	×	×	○
nv	×	×	○
g	×	○	○

$\underline{(0+\mathsf{s}(\bullet))}+\Delta_2 \underset{\mathsf{g}}{\to} \underline{\mathsf{s}(\bullet)+\Delta_2} \underset{\mathsf{g}}{\to} \mathsf{s}(\underline{0+\Delta_2}) \underset{\mathsf{g}}{\to} \mathsf{s}(\underline{\Delta_2}) \overset{*}{\underset{\mathsf{g}}{\to}}$ normal form

TREE AUTOMATA TECHNIQUES

DEFINITION

- bottom-up tree automaton is quadruple $\mathcal{A}=(\mathcal{F},Q,Q_f,\Delta)$ with
  1. $\mathcal{F}$ signature
  2. Q states
  3. $Q_f \subseteq Q$ final states
  4. Δ transition rules $\quad \mathsf{f}(q_1,\ldots,q_n) \to q$
- $L(\mathcal{A})=\{\,t\in\mathcal{T}(\mathcal{F}) \mid \exists q\in Q_f\colon\ t \overset{*}{\underset{\Delta}{\to}} q\,\}$ language accepted by $\mathcal{A}$
- $L\subseteq\mathcal{T}(\mathcal{F})$ is regular if $L=L(\mathcal{A})$ for some tree automaton $\mathcal{A}$

NOTATION

$\Sigma(t)$ set of ground instances of term t

LEMMA

- $\Sigma(t)$ is regular for every linear term t
- $\mathrm{NF}(\mathcal{R})$ is regular for every left-linear TRS $\mathcal{R}$
- following problems are decidable:

  instance: tree automaton $\mathcal{A}$, ground term t
  question: $t\in L(\mathcal{A})$?

  instance: tree automaton $\mathcal{A}$, arbitrary term t
  question: $\Sigma(t)\cap L(\mathcal{A})=\varnothing$?

NOTATION

$(\xrightarrow[\mathcal{R}]{*})[L]$ set of ground terms that rewrite in $\mathcal{R}$ to term in L

DEFINITION

➜ approximation map α assigns approximation $\mathcal{R}_\alpha$ to every TRS $\mathcal{R}$

➜ approximation map α is regularity preserving if

$(\xrightarrow[\mathcal{R}_\alpha]{*})[L]$ is regular $\quad \forall$ regular L, $\forall$ TRS $\mathcal{R}$

THEOREM

$\alpha \in \{\mathsf{s}, \mathsf{nv}, \mathsf{g}, \mathsf{fo}\}$ is regularity preserving

THEOREM

$\forall$ TRS $\mathcal{R}$

$\forall$ regularity preserving α $\quad$ Δ is $\mathcal{R}_\alpha$-needed ? $\quad$ is decidable

$\forall\, C[\Delta]$

DEFINITION

$\mathsf{CBN}_\alpha = \{\, \mathcal{R} \mid$ every reducible term has $\mathcal{R}_\alpha$-needed redex $\}$

THEOREM

$\forall$ left-linear TRS $\mathcal{R}$

$\forall$ regularity preserving α $\quad$ $\mathcal{R} \in \mathsf{CBN}_\alpha$? $\quad$ is decidable

EXAMPLE

$\mathcal{R}$

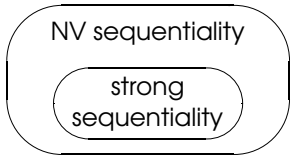
$$0 + y \;\to\; y$$
$$\mathsf{s}(x) + y \;\to\; \mathsf{s}(x + y)$$

term $\quad (\underbrace{0 + \mathsf{s}(\Delta_1)}_{\Delta_3}) + \Delta_2$

redexes	Δ_1	Δ_2	Δ_3
needed	○	○	○
s	×	×	○
nv	×	×	○
g	×	○	○

$\mathcal{R} \in \mathsf{CBN}_\mathsf{s} \subseteq \mathsf{CBN}_\mathsf{nv} \subseteq \mathsf{CBN}_\mathsf{g} \subseteq \mathsf{CBN}_\mathsf{fo}$

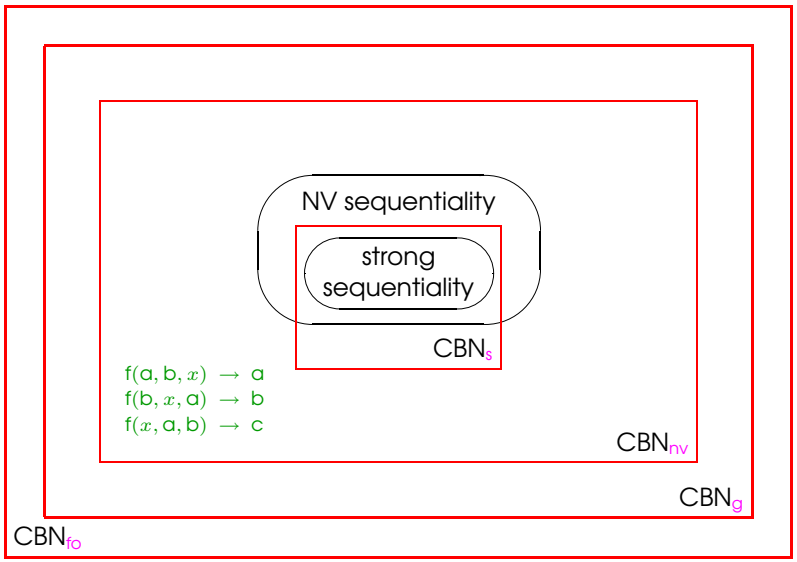
SUMMARY



Oyamaguchi

Huet & Lévy

SUMMARY



REMARK

needed redexes are not independent of other redexes

$$\text{TRS}\quad \begin{array}{rcl} 0+y & \to & y \\ \mathsf{s}(x)+y & \to & \mathsf{s}(x+y) \\ 0\times y & \to & 0 \\ \mathsf{s}(x)\times y & \to & x\times y+y \end{array}$$

redex Δ is needed in $(0+\mathsf{s}(0))\times\Delta$

but not in $(0\times\mathsf{s}(0))\times\Delta$

TERMINATION

DEFINITION

TRS is terminating if there are no infinite rewrite sequences

EXAMPLE

one-rule TRS

$$\mathcal{R}\qquad \mathsf{f}(\mathsf{a},\mathsf{b},x)\ \to\ \mathsf{f}(x,x,x)$$ Toyama '87

is terminating because

$$\neg\,\exists\, s,t\colon\ \mathsf{f}(s,s,s)\ \xrightarrow[\mathcal{R}]{*}\ \mathsf{f}(\mathsf{a},\mathsf{b},t)$$

$$\Longleftrightarrow$$

$$\Sigma(\mathsf{f}(x,x,x))\cap(\xrightarrow[\mathcal{R}]{*})[\Sigma(\mathsf{f}(\mathsf{a},\mathsf{b},x))]=\varnothing$$

$$\Longleftrightarrow$$

$$\Sigma(\mathsf{f}^\sharp(x,x,x))\cap(\xrightarrow[\mathcal{R}]{*})[\Sigma(\mathsf{f}^\sharp(\mathsf{a},\mathsf{b},x))]=\varnothing$$

DEPENDENCY PAIRS

Arts & Giesl TCS'00

DEFINITION

TRS $\mathcal{R}$ over signature $\mathcal{F}$

- ➜ defined symbols $\mathcal{D} = \{\, f \mid f(l_1,\ldots,l_n)\to r\in\mathcal{R}\,\}$
- ➜ marked symbols $\mathcal{D}^\sharp = \{\, f^\sharp \mid f\in\mathcal{D}\,\}$
- ➜ if $t=f(t_1,\ldots,t_n)$ with $f\in\mathcal{D}$ then $t^\sharp=f^\sharp(t_1,\ldots,t_n)$
- ➜ dependency pairs

$$\mathrm{DP}(\mathcal{R}) = \{\, l^\sharp\to t^\sharp \mid l\to r\in\mathcal{R} \text{ and } \underbrace{t\trianglelefteq r}_{t \text{ is subterm of } r} \text{ with } \mathrm{root}(t)\in\mathcal{D}\,\}$$

EXAMPLE

$$\mathcal{R}\quad\begin{array}{lcl} 0-y & \to & 0 \\ x-0 & \to & x \\ \mathsf{s}(x)-\mathsf{s}(y) & \to & x-y \\ 0\div\mathsf{s}(y) & \to & 0 \\ \mathsf{s}(x)\div\mathsf{s}(y) & \to & \mathsf{s}((x-y)\div\mathsf{s}(y)) \end{array}$$

$\mathcal{F}$ 0 s $-$ $\div$

$\mathcal{D}$ $-$ $\div$

$$\mathrm{DP}(\mathcal{R})\quad\begin{array}{lcl} \mathsf{s}(x)-^\sharp\mathsf{s}(y) & \to & x-^\sharp y \\ \mathsf{s}(x)\div^\sharp\mathsf{s}(y) & \to & (x-y)\div^\sharp\mathsf{s}(y) \\ \mathsf{s}(x)\div^\sharp\mathsf{s}(y) & \to & x-^\sharp y \end{array}$$

DEFINITION

reduction pair $(\succsim, >)$ consists of quasi-order $\succsim$ and well-founded order $>$ on terms such that

① $>$ is closed under substitutions
② $\succsim$ is closed under contexts and substitutions
③ $\succsim \cdot > \cdot \succsim \subseteq >$

THEOREM

TRS $\mathcal{R}$ is terminating $\iff$ $\exists$ reduction pair $(\succsim, >)$ such that

① $l \succsim r$ for all $l \to r \in \mathcal{R}$
② $l > r$ for all $l \to r \in \mathrm{DP}(\mathcal{R})$

EXAMPLE

$$\begin{array}{rcl} 0-y & \succsim & 0 \\ x-0 & \succsim & x \\ \mathsf{s}(x)-\mathsf{s}(y) & \succsim & x-y \\ 0\div\mathsf{s}(y) & \succsim & 0 \\ \mathsf{s}(x)\div\mathsf{s}(y) & \succsim & \mathsf{s}((x-y)\div\mathsf{s}(y)) \end{array}$$

$$\begin{array}{rcl} \mathsf{s}(x)-^\sharp\mathsf{s}(y) & > & x-^\sharp y \\ \mathsf{s}(x)\div^\sharp\mathsf{s}(y) & > & (x-y)\div^\sharp\mathsf{s}(y) \\ \mathsf{s}(x)\div^\sharp\mathsf{s}(y) & > & x-^\sharp y \end{array}$$

simplify constraints using argument filtering

DEFINITION

➜ argument filtering (AF) is mapping π such that for every n-ary function symbol f one of following alternatives holds:

① $\pi(f) = i$ with $i \in \{1, \ldots, n\}$
② $\pi(f) = [i_1, \ldots, i_m]$ with $1 \leqslant i_1 \leqslant \cdots \leqslant i_m \leqslant n$

➜ $$\pi(t) = \begin{cases} t & \text{if } t \text{ is variable} \\ \pi(t_i) & \text{if } t = f(t_1, \ldots, t_n) \text{ and ①} \\ f(\pi(t_{i_1}), \ldots, \pi(t_{i_m})) & \text{if } t = f(t_1, \ldots, t_n) \text{ and ②} \end{cases}$$

THEOREM

TRS $\mathcal{R}$ is terminating $\iff$ $\exists$ AF π $\exists$ reduction pair $(\succsim, >)$ such that

① $\pi(l) \succsim \pi(r)$ for all $l \to r \in \mathcal{R}$ $\quad \pi(\mathcal{R}) \subseteq \succsim$
② $\pi(l) > \pi(r)$ for all $l \to r \in \mathrm{DP}(\mathcal{R})$ $\quad \pi(\mathrm{DP}(\mathcal{R})) \subseteq >$

EXAMPLE

$$\begin{array}{rcl} 0 & \succsim & 0 \\ x & \succsim & x \\ \mathsf{s}(x) & \succsim & x \\ 0 \div \mathsf{s}(y) & \succsim & 0 \\ \mathsf{s}(x) \div \mathsf{s}(y) & \succ & \mathsf{s}(\ x \quad \div \mathsf{s}(y)) \end{array}$$

$$\begin{array}{rcl} \mathsf{s}(x) -^\sharp \mathsf{s}(y) & > & x -^\sharp y \\ \mathsf{s}(x) \div^\sharp \mathsf{s}(y) & > & x \quad \div^\sharp \mathsf{s}(y) \\ \mathsf{s}(x) \div^\sharp \mathsf{s}(y) & > & x -^\sharp y \end{array}$$

AF $\quad \pi(-) = 1 \quad \pi(\mathsf{s}) = [1] \quad \pi(0) = [\,]$
$\pi(\div) = \pi(-^\sharp) = \pi(\div^\sharp) = [1,2]$

RPO $\quad \div \sqsupset \mathsf{s} \quad \div^\sharp \sqsupset -^\sharp$

weaken constraints by considering cycles in dependency graph

DG($\mathcal{R}$)
- nodes: $l \to r$ for every $l \to r \in \mathrm{DP}(\mathcal{R})$
- arrows: $l_1 \to r_1 \to l_2 \to r_2$ if $\exists$ substitutions σ, τ such that $r_1\sigma \xrightarrow[\mathcal{R}]{*} l_2\tau$

THEOREM

finite TRS $\mathcal{R}$ is terminating $\iff$ $\forall$ cycle $\mathcal{C}$ in DG($\mathcal{R}$)
$\exists$ AF π $\exists$ reduction pair $(\gtrsim, >)$ such that
① $\pi(\mathcal{R}) \subseteq \gtrsim$
② $\pi(\mathcal{C}) \subseteq \gtrsim \cup >$ and $\pi(\mathcal{C}) \cap > \neq \varnothing$

EXAMPLE

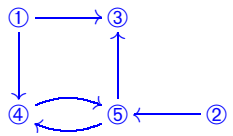
$\mathcal{R}$

$$\begin{array}{lll} \mathsf{even}(x) \to \mathsf{eo}(x,0) & \mathsf{eo}(x,0) \to \mathsf{not}(\mathsf{eo}(x,\mathsf{s}(0))) \\ \mathsf{odd}(x) \to \mathsf{eo}(x,\mathsf{s}(0)) & \mathsf{eo}(0,\mathsf{s}(0)) \to \mathsf{false} \\ \mathsf{not}(\mathsf{true}) \to \mathsf{false} & \mathsf{eo}(\mathsf{s}(x),\mathsf{s}(0)) \to \mathsf{eo}(x,0) \\ \mathsf{not}(\mathsf{false}) \to \mathsf{true} \end{array}$$

DP($\mathcal{R}$)

$$\begin{array}{lll} ① \ \mathsf{even}^\sharp(x) \to \mathsf{eo}^\sharp(x,0) & ④ \ \mathsf{eo}^\sharp(x,0) \to \mathsf{eo}^\sharp(x,\mathsf{s}(0)) \\ ② \ \mathsf{odd}^\sharp(x) \to \mathsf{eo}^\sharp(x,\mathsf{s}(0)) & ⑤ \ \mathsf{eo}^\sharp(\mathsf{s}(x),\mathsf{s}(0)) \to \mathsf{eo}^\sharp(x,0) \\ ③ \ \mathsf{eo}^\sharp(x,0) \to \mathsf{not}^\sharp(\mathsf{eo}(x,\mathsf{s}(0))) \end{array}$$

DG($\mathcal{R}$)



①, ② and ③ don't occur on cycle

EXAMPLE

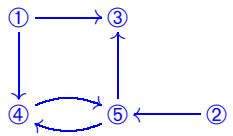
$\mathcal{R}$

$$\begin{array}{lll} \mathsf{even}(x) \to \mathsf{eo} & \mathsf{eo} \to \mathsf{not} \\ \mathsf{odd}(x) \to \mathsf{eo} & \mathsf{eo} \to \mathsf{false} \\ \mathsf{not} \to \mathsf{false} & \mathsf{eo} \to \mathsf{eo} \\ \mathsf{not} \to \mathsf{true} \end{array}$$

DP($\mathcal{R}$)

$$\begin{array}{lll} ④ \ \mathsf{eo}^\sharp(x\ ) \to \mathsf{eo}^\sharp(x\ ) \\ ⑤ \ \mathsf{eo}^\sharp(\mathsf{s}(x)\ ) \to \mathsf{eo}^\sharp(x\ ) \end{array}$$

DG($\mathcal{R}$)



①, ② and ③ don't occur on cycle

AF $\quad \pi(\mathsf{eo}) = \pi(\mathsf{not}) = [\,] \quad \pi(\mathsf{eo}^\sharp) = [1]$

RPO $\quad \begin{array}{l} \mathsf{even} > \\ \mathsf{odd} > \end{array} \mathsf{eo} > \mathsf{not} \begin{array}{l} > \mathsf{false} \\ > \mathsf{true} \end{array}$

weaken constraints by considering cycles in dependency graph

$DG(\mathcal{R})$

nodes: $l \to r$ for every $l \to r \in DP(\mathcal{R})$

arrows: $l_1 \to r_1 \longrightarrow l_2 \to r_2$

if $\exists$ substitutions σ, τ such that $r_1\sigma \xrightarrow[\mathcal{R}]{*} l_2\tau$

THEOREM

finite TRS $\mathcal{R}$ is terminating $\iff$ $\forall$ cycle $\mathcal{C}$ in $DG(\mathcal{R})$

$\exists$ AF π $\exists$ reduction pair $(\gtrsim, >)$ such that

① $\pi(\mathcal{R}) \subseteq \gtrsim$

② $\pi(\mathcal{C}) \subseteq \gtrsim \cup >$ and $\pi(\mathcal{C}) \cap > \neq \varnothing$

PROBLEM

$DG(\mathcal{R})$ is not computable because $r_1\sigma \xrightarrow[\mathcal{R}]{*} l_2\tau$ is undecidable

APPROXIMATING DEPENDENCY GRAPH

DEFINITION Arts & Giesl TCS'00

➜ CAP replace outermost subterms with defined root symbol by distinct fresh variables

➜ REN replace variables by distinct fresh variables

➜ estimated dependency graph

$EDG(\mathcal{R})$

nodes: $l \to r$ for every $l \to r \in DP(\mathcal{R})$

arrows: $l_1 \to r_1 \longrightarrow l_2 \to r_2$

if $REN(CAP(r_1))$ and l_2 are unifiable

LEMMA

➜ $DG(\mathcal{R}) \subseteq EDG(\mathcal{R})$

➜ $EDG(\mathcal{R})$ is computable

EXAMPLE

$\mathcal{R}$ $\quad f(a,b,x) \to f(x,x,x)$

$DP(\mathcal{R})$ $\quad f^\sharp(a,b,x) \to f^\sharp(x,x,x)$ ①

$DG(\mathcal{R})$ ① $\quad \neg\exists\, s,t \quad f^\sharp(s,s,s) \xrightarrow[\mathcal{R}]{*} f^\sharp(a,b,t)$

$EDG(\mathcal{R})$ ①

$REN(CAP(f^\sharp(x,x,x))) = f^\sharp(x_1,x_2,x_3)$ unifies with $f^\sharp(a,b,x)$

DEFINITION

α-approximated dependency graph

$DG_\alpha(\mathcal{R})$

nodes: $l \to r$ for every $l \to r \in DP(\mathcal{R})$

arrows: $l_1 \to r_1 \longrightarrow l_2 \to r_2$

if both $\begin{cases} \Sigma(r_1) \cap (\xrightarrow[\mathcal{R}_\alpha]{*})[\Sigma(REN(l_2))] \neq \varnothing \\ \Sigma(l_2) \cap (\xrightarrow[(\mathcal{R}^{-1})_\alpha]{*})[\Sigma(REN(r_1))] \neq \varnothing \end{cases}$

$\left(DG(\mathcal{R}): \quad \Sigma(r_1) \cap (\xrightarrow[\mathcal{R}]{*})[\Sigma(l_2)] \neq \varnothing \iff \Sigma(l_2) \cap (\xrightarrow[\mathcal{R}^{-1}]{*})[\Sigma(r_1)] \neq \varnothing\right)$

LEMMA $\forall\, \alpha \in \{s, nv, g, fo\}$

➜ $DG(\mathcal{R}) \subseteq DG_\alpha(\mathcal{R})$

➜ $DG_\alpha(\mathcal{R})$ is computable

THEOREM

- ➜ $DG_{fo}(\mathcal{R}) \subseteq DG_{g}(\mathcal{R}) \subseteq DG_{nv}(\mathcal{R}) \subseteq DG_{s}(\mathcal{R})$
- ➜ $DG_{s}(\mathcal{R}) \subseteq EDG(\mathcal{R})$ for every left-linear $\mathcal{R}$
- ➜ $DG_{nv}(\mathcal{R}) \subseteq EDG(\mathcal{R})$

EXAMPLE

$\mathcal{R}$ $\quad f(a,b,x) \to f(x,x,x) \qquad f(a,b,x) \to f(x_1,x_2,x_3)$ $\quad \mathcal{R}_{nv}$

$DP(\mathcal{R})$ $\quad f^\sharp(a,b,x) \to f^\sharp(x,x,x)$ ①

$EDG(\mathcal{R})$ ①

$DG_{nv}(\mathcal{R})$ ①

$$(\xrightarrow[\mathcal{R}_{nv}]{*})[\Sigma(f^\sharp(a,b,x))] = \Sigma(f^\sharp(a,b,x)) \ \wedge \ \Sigma(f^\sharp(a,b,x)) \cap \Sigma(f^\sharp(x,x,x)) = \varnothing$$

THREE REMARKS

$\mathcal{R}$ $\quad f(a,b,x) \to f(x,x,x)$

① $\mathcal{R}$ is not DP quasi-simply terminating

> "DP quasi-simple termination captures all TRSs where an automated termination proof using dependency pairs is potentially feasible"
>
> Giesl & Ohlebusch FROCOS'98

② recent refinements of dependency pair method (narrowing, rewriting, instantiation) are not applicable to $\mathcal{R}$

Giesl & Arts AAECC'01

③ nv-approximated dependency graph is (much) better than Kusakari and Toyama's approximation based on Ω-reduction

Kusakari & Toyama '98, '00

NEW APPROXIMATION

improve estimated dependency graph by symmetry

DEFINITION

- ➜ $\mathcal{D}^{-1} = \mathcal{D}(\mathcal{R}^{-1})$ $\quad$ (if $\mathcal{R}$ is collapsing then $\mathcal{D}^{-1} = \mathcal{F}$)
- ➜ CAP^{-1} replace outermost subterms with root symbol in $\mathcal{D}^{-1}$ by fresh variables
- ➜ estimated* dependency graph

$EDG^*(\mathcal{R})$

nodes: $\boxed{l \to r}$ for every $l \to r \in DP(\mathcal{R})$

arrows: $\boxed{l_1 \to r_1} \longrightarrow \boxed{l_2 \to r_2}$

if both $\begin{cases} REN(CAP(r_1)) \text{ and } l_2 \text{ are unifiable} \\ r_1 \text{ and } REN(CAP^{-1}(l_2)) \text{ are unifiable} \end{cases}$

EXAMPLE

$\mathcal{R}$ $\quad f(a,b,x) \to f(x,x,x)$

$DP(\mathcal{R})$ $\quad f^\sharp(a,b,x) \to f^\sharp(x,x,x)$ ①

$DG(\mathcal{R})$ ① $\qquad \neg\exists\, s,t \quad f^\sharp(s,s,s) \xrightarrow[\mathcal{R}]{*} f^\sharp(a,b,t)$

$EDG(\mathcal{R})$ ①

$REN(CAP(f^\sharp(x,x,x))) = f^\sharp(x_1,x_2,x_3)$ unifies with $f^\sharp(a,b,x)$

$EDG^*(\mathcal{R})$ ①

$REN(CAP^{-1}(f^\sharp(a,b,x))) = f^\sharp(a,b,x_1)$ doesn't unify with $f^\sharp(x,x,x)$

SUMMARY