Advanced Topics in Termination

ISR 2009 – lecture 2

Aart Middeldorp

Institute of Computer Science
University of Innsbruck

Topics

- polynomial interpretations
- matrix interpretations
- dependency pairs
- match-bounds
- semantic labeling
- derivational complexity
- decidable classes
- certification
- applications

Further Reading

- http://cl-informatik.uibk.ac.at/~ami/09isr/

Puzzle

$$\mathsf{f}(\mathsf{0}) \to \mathsf{0} \qquad \mathsf{s}(\mathsf{s}(\mathsf{0})) \to \mathsf{f}(\mathsf{0})$$
$$\mathsf{f}(\mathsf{s}(\mathsf{0})) \to \mathsf{s}(\mathsf{0}) \qquad \mathsf{s}(\mathsf{s}(\mathsf{s}(\mathsf{0}))) \to \mathsf{f}(\mathsf{s}(\mathsf{0}))$$
$$\mathsf{f}(\mathsf{s}(\mathsf{s}(\mathsf{0}))) \to \mathsf{s}(\mathsf{s}(\mathsf{s}(\mathsf{s}(\mathsf{s}(\mathsf{0}))))) \qquad \mathsf{s}(\mathsf{s}(\mathsf{s}(\mathsf{s}(\mathsf{s}(\mathsf{s}(\mathsf{0})))))) \to \mathsf{f}(\mathsf{s}(\mathsf{s}(\mathsf{0})))$$

polynomially terminating over $\mathbb{N}$?

Solutions

- $\mathsf{f}_{\mathbb{N}}(x) = 2x^3 + 2x^2 + x + 1 \qquad \mathsf{s}_{\mathbb{N}}(x) = 2x + 1 \qquad \mathsf{0}_{\mathbb{N}} = 0$
- $\mathsf{f}_{\mathbb{N}}(x) = 2x^2 - x + 1 \qquad \mathsf{s}_{\mathbb{N}}(x) = x + 1 \qquad \mathsf{0}_{\mathbb{N}} = 0$

Outline

Lemma

$\forall$ non-terminating TRS $\exists$ minimal non-terminating term

all proper subterms are terminating

Notation

$\mathcal{T}_\infty$ is set of all minimal non-terminating terms

Lemma

$\forall\, t \in \mathcal{T}_\infty \quad \exists\, l \to r \in \mathcal{R} \quad \exists\, \sigma \quad \exists$ non-variable subterm u of r

$$t \xrightarrow{>\epsilon}{}^{*} l\sigma \xrightarrow{\epsilon} r\sigma \trianglerighteq u\sigma \in \mathcal{T}_\infty$$

Corollary

every term in $\mathcal{T}_\infty$ has defined root symbol

Lemma

$\forall\, t \in \mathcal{T}_\infty \quad \exists\, l \to r \in \mathcal{R} \quad \exists\, \sigma \quad \exists$ non-variable subterm u of r with $u \ntrianglelefteq l$

$$t \xrightarrow{>\epsilon}{}^{*} l\sigma \xrightarrow{\epsilon} r\sigma \trianglerighteq u\sigma \in \mathcal{T}_\infty$$

(Tentative) Definition

$\mathcal{S} = \{l \to u \mid l \to r \in \mathcal{R},\ u \trianglelefteq r$ with $\mathsf{root}(u)$ defined, $u \ntrianglelefteq l\}$

Lemma

$\forall\, t \in \mathcal{T}_\infty \quad \exists\, l \to u \in \mathcal{S} \quad \exists\, \sigma \quad t \xrightarrow[\mathcal{R}]{>\epsilon}{}^{*} l\sigma \xrightarrow[\mathcal{S}]{\epsilon} u\sigma \in \mathcal{T}_\infty$

Idea

get rid of position constraints by marking root symbols of terms in rules of $\mathcal{S}$

Definition

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

- $\mathcal{F}^\sharp = \mathcal{F} \cup \{f^\sharp \mid f$ is defined symbol of $\mathcal{R}\}$
- if $t = f(t_1,\dots,t_n)$ with f defined then $t^\sharp = f^\sharp(t_1,\dots,t_n)$
- $\mathcal{T}_\infty^\sharp = \{t^\sharp \mid t \in \mathcal{T}_\infty\}$
- DP($\mathcal{R}$) $= \{l^\sharp \to u^\sharp \mid l \to r \in \mathcal{R},\ u \trianglelefteq r$ with $\mathsf{root}(u)$ defined, $u \ntrianglelefteq l\}$

dependency pair

Lemma

$\forall\, s \in \mathcal{T}_\infty \quad \exists\, t, u \in \mathcal{T}_\infty \qquad s^\sharp \xrightarrow[\mathcal{R}]{*} t^\sharp \xrightarrow[DP(\mathcal{R})]{} u^\sharp$

Corollary

$\forall$ non-terminating TRS $\mathcal{R}$ $\exists$ infinite rewrite sequence

$$\mathcal{T}_\infty^\sharp \ni t_1 \xrightarrow[\mathcal{R}]{*} t_2 \xrightarrow[DP(\mathcal{R})]{} t_3 \xrightarrow[\mathcal{R}]{*} t_4 \xrightarrow[DP(\mathcal{R})]{} \cdots$$

Example

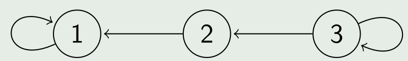
rewrite rules

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

dependency pairs

$$1:\quad \mathsf{s}(x) +^\sharp y \to x +^\sharp y$$
$$2:\quad \mathsf{s}(x) \times^\sharp y \to (x \times y) +^\sharp y$$
$$3:\quad \mathsf{s}(x) \times^\sharp y \to x \times^\sharp y$$

dependency graph



strongly connected components

$$\{1\} \qquad \{3\}$$

Outline

- Dependency Pairs
- **Dependency Graph**
- Processors
- Subterm Criterion
- Reduction Pairs
- Argument Filters
- Usable Rules
- Further Reading

Corollary

$\forall$ finite non-terminating TRS $\mathcal{R}$ $\exists$ infinite rewrite sequence

$$\mathcal{T}^{\sharp}_{\infty} \ni t_1 \xrightarrow[\mathcal{R}]{*} t_2 \xrightarrow[\mathcal{P}]{} t_3 \xrightarrow[\mathcal{R}]{*} t_4 \xrightarrow[\mathcal{P}]{} \cdots$$

with $\mathcal{P} \subseteq DP(\mathcal{R})$

Observation

$\mathcal{P}$ is strongly connected component in dependency graph of $\mathcal{R}$

Definition

dependency graph $\mathrm{DG}(\mathcal{R})$ of TRS $\mathcal{R}$ consists of

- nodes: dependency pairs of $\mathcal{R}$
- arrows: $s \to t \longrightarrow u \to v$ if $\exists$ substitutions σ, τ such that $t\sigma \xrightarrow[\mathcal{R}]{*} u\tau$

Remark

dependency graph is not computable but good over-approximations exist

Trivial Approximation

$$s \to t \xrightarrow{t} u \to v \quad \iff \quad \mathrm{root}(t) = \mathrm{root}(u)$$

Better Approximations

are based on

- unification ▸ details
- tree automata techniques

Example

rewrite rules

$$0 + y \to y \qquad 0 \times y \to 0$$

$$\mathsf{s}(x) + y \to \mathsf{s}(x + y) \qquad \mathsf{s}(x) \times y \to (x \times y) + y$$

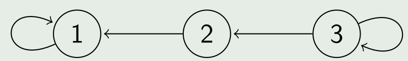
dependency pairs

1: $\mathsf{s}(x) +^{\sharp} y \to x +^{\sharp} y$

2: $\mathsf{s}(x) \times^{\sharp} y \to (x \times y) +^{\sharp} y$

3: $\mathsf{s}(x) \times^{\sharp} y \to x \times^{\sharp} y$

dependency graph



strongly connected components

$$\{1\} \qquad \{3\}$$

Outline

Definition

DP problem is pair of TRSs $(\mathcal{P}, \mathcal{R})$ such that root symbols of rules in $\mathcal{P}$

- do not occur in $\mathcal{R}$
- do not occur in proper subterms of left- and right-hand sides of rules in $\mathcal{P}$

Example

$(\mathsf{DP}(\mathcal{R}), \mathcal{R})$ is DP problem for every TRS $\mathcal{R}$

Definition

DP problem $(\mathcal{P}, \mathcal{R})$ is finite if $\neg\,\exists$ infinite sequence

$$t_1 \xrightarrow[\mathcal{R}]{*} t_2 \xrightarrow[\mathcal{P}]{\epsilon} t_3 \xrightarrow[\mathcal{R}]{*} t_4 \xrightarrow[\mathcal{P}]{\epsilon} \cdots$$

such that all terms t_1, t_2, ... are terminating with respect to $\mathcal{R}$

Theorem

TRS $\mathcal{R}$ is terminating $\iff$ *DP problem $(\mathsf{DP}(\mathcal{R}), \mathcal{R})$ is finite*

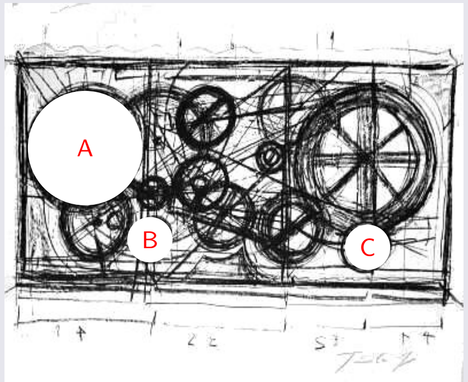
Definition

- DP processor is function from DP problems to sets of DP problems
- DP processor Φ is sound if DP problem $(\mathcal{P}, \mathcal{R})$ is finite whenever all DP problems in $\Phi(\mathcal{P}, \mathcal{R})$ are finite
- DP processor Φ is complete if DP problem $(\mathcal{P}, \mathcal{R})$ is not finite whenever at least one DP problem in $\Phi(\mathcal{P}, \mathcal{R})$ is not finite

DP Processors

dependency graph, increasing interpretations, instantiation, loop detection, match-bounds, narrowing, reduction pairs, rewriting, root-labeling, rule removal, size-change principle, semantic labeling, string reversal, subterm criterion, uncurrying, usable rules, ...

Termination Tools



DP processors

Definition

dependency graph $\mathrm{DG}(\mathcal{P},\mathcal{R})$ of DP problem $(\mathcal{P},\mathcal{R})$ consists of

- nodes: rules in $\mathcal{P}$
- arrows: $s \to t \longrightarrow u \to v$ if $\exists$ substitutions σ, τ such that $t\sigma \xrightarrow[\mathcal{R}]{*} u\tau$

Definition (Dependency Graph Processor)

$(\mathcal{P},\mathcal{R}) \mapsto \{(\mathcal{S},\mathcal{R}) \mid \mathcal{S} \text{ is strongly connected component in } \mathrm{DG}(\mathcal{P},\mathcal{R})\}$

Theorem

dependency graph processor is sound and complete

Outline

Definition

DP problem $(\mathcal{P},\mathcal{R})$ is finite if $\neg\,\exists$ infinite sequence

$$t_1 \xrightarrow[\mathcal{R}]{*} t_2 \xrightarrow[\mathcal{P}]{\epsilon} t_3 \xrightarrow[\mathcal{R}]{*} t_4 \xrightarrow[\mathcal{P}]{\epsilon} \cdots$$

such that all terms t_1, t_2, ... are terminating with respect to $\mathcal{R}$

Question

how to prove finiteness of DP problem $(\mathcal{P},\mathcal{R})$?

Answers

- use pair of orders
- use minimality together with subterm relation
- ...

Idea

project each dependency pair symbol in $\mathcal{P}$ to fixed argument position

$$\pi(t_1) \xrightarrow[\mathcal{R}]{*} \pi(t_2) \quad ? \quad \pi(t_3) \xrightarrow[\mathcal{R}]{*} \pi(t_4) \quad ? \quad \cdots$$

Observation

$\pi(t_1)$ is terminating with respect to $\xrightarrow[\mathcal{R}]{} \cup \rhd$

Definition

- simple projection for DP problem $(\mathcal{P},\mathcal{R})$ is mapping π that assigns to every dependency pair symbol $f^\sharp$ in $\mathcal{P}$ one of its argument positions
- extension to terms in $\mathcal{T}^\sharp$: $\pi(f^\sharp(t_1,\dots,t_n)) = t_{\pi(f^\sharp)}$

Definition (Subterm Criterion Processor)

simple projection π

$$(\mathcal{P},\mathcal{R}) \mapsto \begin{cases} \{(\{l \to r \in \mathcal{P} \mid \pi(l) = \pi(r)\}, \mathcal{R})\} & \text{if } \pi(\mathcal{P}) \subseteq \trianglerighteq \\ \{(\mathcal{P},\mathcal{R})\} & \text{otherwise} \end{cases}$$

Theorem

subterm criterion processor is sound and complete

Example

rewrite rules

$$\mathsf{ack}(0, y) \to \mathsf{s}(y)$$
$$\mathsf{ack}(\mathsf{s}(x), 0) \to \mathsf{ack}(x, \mathsf{s}(0))$$
$$\mathsf{ack}(\mathsf{s}(x), \mathsf{s}(y)) \to \mathsf{ack}(x, \mathsf{ack}(\mathsf{s}(x), y))$$

dependency pairs

$$\mathsf{ack}^\sharp(\mathsf{s}(x), 0) \to \mathsf{ack}^\sharp(x, \mathsf{s}(0))$$
$$\mathsf{ack}^\sharp(\mathsf{s}(x), \mathsf{s}(y)) \to \mathsf{ack}^\sharp(x, \mathsf{ack}(\mathsf{s}(x), y))$$
$$\mathsf{ack}^\sharp(\mathsf{s}(x), \mathsf{s}(y)) \to \mathsf{ack}^\sharp(\mathsf{s}(x), y)$$

simple projection

$$\pi(\mathsf{ack}^\sharp) = 1$$

Example

rewrite rules

$$\mathsf{ack}(0, y) \to \mathsf{s}(y)$$
$$\mathsf{ack}(\mathsf{s}(x), 0) \to \mathsf{ack}(x, \mathsf{s}(0))$$
$$\mathsf{ack}(\mathsf{s}(x), \mathsf{s}(y)) \to \mathsf{ack}(x, \mathsf{ack}(\mathsf{s}(x), y))$$

dependency pairs

$$\mathsf{ack}^\sharp(\mathsf{s}(x), \mathsf{s}(y)) \to \mathsf{ack}^\sharp(\mathsf{s}(x), y)$$

simple projection

$$\pi(\mathsf{ack}^\sharp) = 2$$

Outline

Definition

DP problem $(\mathcal{P}, \mathcal{R})$ is finite if $\neg\,\exists$ infinite sequence

$$t_1 \gtrsim t_2 > t_3 \gtrsim t_4 > \cdots$$

such that all terms t_1, t_2, ... are terminating with respect to $\mathcal{R}$

Question

how to prove finiteness of DP problem $(\mathcal{P}, \mathcal{R})$?

Answers

- use pair of orders $\gtrsim$ and $>$ such that $\mathcal{R} \subseteq \gtrsim$ and $\mathcal{P} \subseteq >$
- use minimality together with subterm relation
- ...

Definition

reduction pair $(>, \gtrsim)$ consists of well-founded order $>$ and preorder $\gtrsim$ such that

1. $>$ is closed under substitutions
2. $\gtrsim$ is closed under contexts and substitutions
3. $> \cdot \gtrsim \subseteq >$ or $\gtrsim \cdot > \subseteq >$

Definition (Reduction Pair Processor)

reduction pair $(>, \gtrsim)$

$$(\mathcal{P}, \mathcal{R}) \mapsto \begin{cases} \{(\mathcal{P} \setminus >, \mathcal{R})\} & \text{if } \mathcal{P} \subseteq > \cup \gtrsim \text{ and } \mathcal{R} \subseteq \gtrsim \\ \{(\mathcal{P}, \mathcal{R})\} & \text{otherwise} \end{cases}$$

Theorem

reduction pair processor is sound and complete

Definitions

- well-founded monotone $\mathcal{F}$-algebra $(\mathcal{A}, >)$ consists of nonempty algebra $\mathcal{A} = (A, \{f_\mathcal{A}\}_{f \in \mathcal{F}})$ together with well-founded order $>$ on A such that every $f_\mathcal{A}$ is strictly monotone in all coordinates:

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

  for all $a_1, \ldots, a_n, b \in A$ and $i \in \{1, \ldots, n\}$ with $a_i > b$
- relation $>_\mathcal{A}$ on terms: $s >_\mathcal{A} t$ if $[\alpha]_\mathcal{A}(s) > [\alpha]_\mathcal{A}(t)$ for all assignments α
- relation $\geqslant_\mathcal{A}$ on terms: $s \geqslant_\mathcal{A} t$ if $[\alpha]_\mathcal{A}(s) \geqslant [\alpha]_\mathcal{A}(t)$ for all assignments α

Lemma

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

Remark

$>_\mathcal{A}$ is closed under contexts for every well-founded monotone algebra $\mathcal{A}$

Definition

monotonic reduction pair is reduction pair $(>, \gtrsim)$ with $>$ closed under contexts

Definition (Rule Removal Processor)

monotonic reduction pair $(>, \gtrsim)$

$$(\mathcal{P}, \mathcal{R}) \mapsto \begin{cases} \{(\mathcal{P} \setminus >, \mathcal{R} \setminus >)\} & \text{if } \mathcal{P} \subseteq > \cup \gtrsim \text{ and } \mathcal{R} \subseteq \gtrsim \cup > \\ \{(\mathcal{P}, \mathcal{R})\} & \text{otherwise} \end{cases}$$

Theorem

rule removal processor is sound and complete

Definition

weakly monotone $\mathcal{F}$-algebra $(\mathcal{A}, >, \gtrsim)$ is nonempty algebra $\mathcal{A} = (A, \{f_\mathcal{A}\}_{f \in \mathcal{F}})$ together with well-founded order $>$ and preorder $\gtrsim$ such that

1. $f_\mathcal{A}$ is monotone with respect to $\gtrsim$ in all coordinates
2. $> \cdot \gtrsim \subseteq >$ or $\gtrsim \cdot > \subseteq >$

Lemma

if $(\mathcal{A}, >, \gtrsim)$ is weakly monotone algebra then $(>_\mathcal{A}, \gtrsim_\mathcal{A})$ is reduction pair

Example

- $A = \mathbb{N}$
- $\gtrsim \, = \, \geqslant$
- $f_\mathcal{A} \in \mathbb{Z}[x_1, \ldots, x_n]$ for every n-ary function symbol f

Example

rewrite rules

$$\mathsf{while}(\mathsf{true}, x, y) \to \mathsf{while}(x > y, \mathsf{p}(x), y) \qquad 0 > y \to \mathsf{false}$$
$$\mathsf{p}(0) \to 0 \qquad \mathsf{s}(x) > 0 \to \mathsf{true}$$
$$\mathsf{p}(\mathsf{s}(x)) \to x \qquad \mathsf{s}(x) > \mathsf{s}(y) \to x > y$$

dependency pairs

$$\mathsf{while}^\sharp(\mathsf{true}, x, y) \to \mathsf{while}^\sharp(x > y, \mathsf{p}(x), y) \qquad \mathsf{s}(x) >^\sharp \mathsf{s}(y) \to x >^\sharp y$$
$$\mathsf{while}^\sharp(\mathsf{true}, x, y) \to x >^\sharp y \qquad \mathsf{while}^\sharp(\mathsf{true}, x, y) \to \mathsf{p}^\sharp(x)$$

interpretations

$$\mathsf{while}_\mathbb{N}(x, y, z) = >_\mathbb{N}(x, y) = \mathsf{true}_\mathbb{N} = \mathsf{false}_\mathbb{N} = 0_\mathbb{N} = 0$$
$$\mathsf{while}^\sharp_\mathbb{N}(x, y, z) = y + 1 \qquad \mathsf{s}_\mathbb{N}(x) = x + 1 \qquad >^\sharp_\mathbb{N}(x, y) = \mathsf{p}_\mathbb{N}(x) = \mathsf{p}^\sharp_\mathbb{N}(x) = x$$

Example

rewrite rules

$$\mathsf{while}(\mathsf{true}, x, y) \to \mathsf{while}(x > y, \mathsf{p}(x), y) \qquad 0 > y \to \mathsf{false}$$
$$\mathsf{p}(0) \to 0 \qquad \mathsf{s}(x) > 0 \to \mathsf{true}$$
$$\mathsf{p}(\mathsf{s}(x)) \to x \qquad \mathsf{s}(x) > \mathsf{s}(y) \to x > y$$

dependency pairs

$$\mathsf{while}^\sharp(\mathsf{true}, x, y) \to \mathsf{while}^\sharp(x > y, \mathsf{p}(x), y)$$

interpretations

$$\mathsf{while}_\mathbb{Q}(x, y, z) = \mathsf{false}_\mathbb{Q} = 0 \qquad >_\mathbb{Q}(x, y) = x + \tfrac{1}{2} \qquad \mathsf{true}_\mathbb{Q} = \tfrac{7}{2} \qquad 0_\mathbb{Q} = 1$$
$$\mathsf{while}^\sharp_\mathbb{Q}(x, y, z) = x + \tfrac{5}{2}y \qquad \mathsf{s}_\mathbb{Q}(x) = 2x + \tfrac{7}{2} \qquad \mathsf{p}_\mathbb{Q}(x) = \tfrac{1}{2}x + 1$$

Definition

$\mathbb{A}_\mathbb{N} = \mathbb{N} \cup \{-\infty\}$ with

- orders $\cdots > 1 > 0 > -\infty$ and $\gg \, = \, > \cup \{(-\infty, -\infty)\}$
- operations
  - $x \oplus y = \max(x, y)$
  - $x \otimes y = x + y \qquad -\infty \otimes x = x \otimes -\infty = -\infty$

Lemma

if $\mathcal{M}$ is matrix interpretation with carrier $\mathbb{N} \times \mathbb{A}_\mathbb{N}^{d-1}$ and interpretations

$$f_\mathcal{M}(\vec{x}_1, \ldots, \vec{x}_n) = M_1 \otimes \vec{x}_1 \oplus \cdots \oplus M_n \otimes \vec{x}_n \oplus \vec{f}$$

such that $f_1 \geqslant 0$ or $\exists\, i\ (M_i)_{1,1} \geqslant 0$ then $(\mathcal{M}, \gg, \geqslant)$ is weakly monotone algebra

Definition

$\mathbb{A}_{\mathbb{Z}} = \mathbb{Z} \cup \{-\infty\}$ with

- orders $\cdots > 1 > 0 > -1 > \cdots > -\infty$ and $\gg\, = \,> \cup \{(-\infty, -\infty)\}$
- operations
  - $x \oplus y = \max(x, y)$
  - $x \otimes y = x + y \qquad -\infty \otimes x = x \otimes -\infty = -\infty$

Lemma

if $\mathcal{M}$ is matrix interpretation with carrier $\mathbb{N} \times \mathbb{A}_{\mathbb{Z}}^{d-1}$ and interpretations

$$f_{\mathcal{M}}(\vec{x}_1, \ldots, \vec{x}_n) = M_1 \otimes \vec{x}_1 \oplus \cdots \oplus M_n \otimes \vec{x}_n \oplus \vec{f}$$

such that $f_1 \geqslant 0$ then $(\mathcal{M}, \gg, \geqslant)$ is weakly monotone algebra

Example

rewrite rules

$$\begin{aligned} \mathsf{while}(\mathsf{true}, x, y) &\to \mathsf{while}(x > y, \mathsf{p}(x), y) & 0 > y &\to \mathsf{false} \\ \mathsf{p}(0) &\to 0 & \mathsf{s}(x) > 0 &\to \mathsf{true} \\ \mathsf{p}(\mathsf{s}(x)) &\to x & \mathsf{s}(x) > \mathsf{s}(y) &\to x > y \end{aligned}$$

dependency graph analysis

$$\mathsf{while}^\sharp(\mathsf{true}, x, y) \to \mathsf{while}^\sharp(x > y, \mathsf{p}(x), y) \qquad \mathsf{s}(x) >^\sharp \mathsf{s}(y) \to x >^\sharp y$$

$$\mathsf{while}^\sharp(\mathsf{true}, x, y) \to x >^\sharp y \qquad \mathsf{while}^\sharp(\mathsf{true}, x, y) \to \mathsf{p}^\sharp(x)$$

subterm criterion

Example

rewrite rules

$$\begin{aligned} \mathsf{while}(\mathsf{true}, x, y) &\to \mathsf{while}(x > y, \mathsf{p}(x), y) & 0 > y &\to \mathsf{false} \\ \mathsf{p}(0) &\to 0 & \mathsf{s}(x) > 0 &\to \mathsf{true} \\ \mathsf{p}(\mathsf{s}(x)) &\to x & \mathsf{s}(x) > \mathsf{s}(y) &\to x > y \end{aligned}$$

dependency graph analysis

$$\mathsf{while}^\sharp(\mathsf{true}, x, y) \to \mathsf{while}^\sharp(x > y, \mathsf{p}(x), y)$$

arctic interpretation

$$\begin{aligned} \mathsf{while}^\sharp_{\mathcal{M}}(x, y, z) &= x \oplus y \oplus (0) & \mathsf{s}_{\mathbb{N}}(x) &= (2) \otimes x \oplus (3) \\ \mathsf{while}_{\mathcal{M}}(x, y, z) &= x \oplus (2) \otimes y \oplus (0) & \mathsf{p}_{\mathbb{N}}(x) &= (-1) \otimes x \oplus (0) \\ >_{\mathcal{M}}(x, y) &= (-1) \otimes x \oplus (0) & \mathsf{true}_{\mathcal{M}} &= (2) \quad \mathsf{false}_{\mathcal{M}} = 0_{\mathcal{M}} = (0) \end{aligned}$$

Outline

Remark

traditional simplification orders like LPO and KBO give rise to monotonic reduction pairs

Definition

- argument filter is mapping π such that for every n-ary function symbol f

$$\pi(f) \in \{1,\dots,n\} \quad \text{or} \quad \pi(f) = [i_1,\dots,i_m] \text{ with } 1 \leqslant i_1 < \cdots < i_m \leqslant n$$

- extension to terms:

$$\pi(t) = \begin{cases} t & \text{if } t \text{ is variable} \\ \pi(t_i) & \text{if } t = f(t_1,\dots,t_n) \text{ and } \pi(f) = i \\ f(\pi(t_{i_1}),\dots,\pi(t_{i_m})) & \text{if } t = f(t_1,\dots,t_n) \text{ and } \pi(f) = [i_1,\dots,i_m] \end{cases}$$

Notation

- $s >_\pi t \iff \pi(s) > \pi(t)$
- $s \gtrsim_\pi t \iff \pi(s) \gtrsim \pi(t)$

Lemma

$(>_\pi, \gtrsim_\pi)$ is reduction pair for every reduction pair $(>, \gtrsim)$ and argument filter π

Example

DP problem

$$\begin{array}{cc} 0 - y \to 0 & 0 \div \mathsf{s}(y) \to 0 \\ x - 0 \to x & \mathsf{s}(x) \div \mathsf{s}(y) \to \mathsf{s}((x - y) \div \mathsf{s}(y)) \\ \mathsf{s}(x) - \mathsf{s}(y) \to x - y & \mathsf{s}(x) \div^\sharp \mathsf{s}(y) \to (x - y) \div^\sharp \mathsf{s}(y) \end{array}$$

argument filter $\pi(-) = 1$ LPO with precedence $\div > \mathsf{s}$

Outline

Definition (Reduction Pair Processor with Usable Rules)

reduction pair $(>, \gtrsim)$ which is $\mathcal{C}_\mathcal{E}$-compatible

$$(\mathcal{P}, \mathcal{R}) \mapsto \begin{cases} \{(\mathcal{P} \setminus >, \mathcal{R})\} & \text{if } \mathcal{P} \subseteq {>} \cup {\gtrsim} \text{ and } \mathcal{U}(\mathcal{P}, \mathcal{R}) \subseteq {\gtrsim} \\ \{(\mathcal{P}, \mathcal{R})\} & \text{otherwise} \end{cases}$$

Definition

reduction pair $(>, \gtrsim)$ is $\mathcal{C}_\mathcal{E}$-compatible if $\mathsf{c}(x,y) \gtrsim x, y$ for fresh symbol c

Theorem

reduction pair processor with usable rules is sound and complete

Definition

DP problem $(\mathcal{P}, \mathcal{R})$

- if t is variable then $\mathcal{US}(t) = \varnothing$
- if $t = f(t_1, \ldots, t_n)$ then $\mathcal{US}(t)$ is least set such that
  1. $f \in \mathcal{US}(t)$
  2. $\mathcal{US}(t_1) \cup \cdots \cup \mathcal{US}(t_n) \subseteq \mathcal{US}(t)$
  3. if $l \to r \in \mathcal{R}$ and $\mathrm{root}(l) \in \mathcal{US}(t)$ then $\mathcal{F}\mathrm{un}(r) \subseteq \mathcal{US}(t)$
- usable symbols:

$$\mathcal{US}(\mathcal{P}, \mathcal{R}) = \bigcup_{l \to r \in \mathcal{P}} \mathcal{US}(r)$$

- usable rules:

$$\mathcal{U}(\mathcal{P}, \mathcal{R}) = \{ l \to r \in \mathcal{R} \mid \mathrm{root}(l) \in \mathcal{US}(\mathcal{P}, \mathcal{R}) \}$$

Definition (Reduction Pair Processor with Usable Rules)

reduction pair $(>, \gtrsim)$ which is $\mathcal{C}_\varepsilon$-compatible

$$(\mathcal{P}, \mathcal{R}) \mapsto \begin{cases} \{(\mathcal{P} \setminus >, \mathcal{R})\} & \text{if } \mathcal{P} \subseteq > \cup \gtrsim \text{ and } \mathcal{U}(\mathcal{P}, \mathcal{R}) \subseteq \gtrsim \\ \{(\mathcal{P}, \mathcal{R})\} & \text{otherwise} \end{cases}$$

Definition

reduction pair $(>, \gtrsim)$ is $\mathcal{C}_\varepsilon$-compatible if $\mathsf{c}(x, y) \gtrsim x, y$ for fresh symbol c

Theorem

reduction pair processor with usable rules is sound and complete

Puzzle

$\mathcal{C}_\varepsilon$-compatible reduction pair $(>, \gtrsim)$

$$(\mathcal{P}, \mathcal{R}) \mapsto \begin{cases} \{(\mathcal{P} \setminus >, \mathcal{U}(\mathcal{P}, \mathcal{R}))\} & \text{if } \mathcal{P} \subseteq > \cup \gtrsim \text{ and } \mathcal{U}(\mathcal{P}, \mathcal{R}) \subseteq \gtrsim \\ \{(\mathcal{P}, \mathcal{R})\} & \text{otherwise} \end{cases}$$

is this DP processor sound ?

Definition

DP problem $(\mathcal{P}, \mathcal{R})$

- if t is variable then $\mathcal{US}_\pi(t) = \varnothing$
- if $t = f(t_1, \ldots, t_n)$ then $\mathcal{US}_\pi(t)$ is least set such that
  1. $f \in \mathcal{US}_\pi(t)$
  2. if $i \in \pi(f)$ then $\mathcal{US}_\pi(t_i) \subseteq \mathcal{US}_\pi(t)$
  3. if $l \to r \in \mathcal{R}$ and $\mathrm{root}(l) \in \mathcal{US}_\pi(t)$ then $\mathcal{F}\mathrm{un}(r) \subseteq \mathcal{US}_\pi(t)$
- usable symbols with respect to argument filter π:

$$\mathcal{US}_\pi(\mathcal{P}, \mathcal{R}) = \bigcup_{l \to r \in \mathcal{P}} \mathcal{US}_\pi(r)$$

- usable rules with respect to argument filter π:

$$\mathcal{U}_\pi(\mathcal{P}, \mathcal{R}) = \{ l \to r \in \mathcal{R} \mid \mathrm{root}(l) \in \mathcal{US}_\pi(\mathcal{P}, \mathcal{R}) \}$$

Definition (RP Processor with Usable Rules and Argument Filters)

$\mathcal{C}_{\mathcal{E}}$-compatible reduction pair $(>, \gtrsim)$ argument filter π

$$(\mathcal{P}, \mathcal{R}) \mapsto \begin{cases} \{(\mathcal{P} \setminus >_\pi, \mathcal{R})\} & \text{if } \mathcal{P} \subseteq >_\pi \cup \gtrsim_\pi \text{ and } \mathcal{U}_\pi(\mathcal{P}, \mathcal{R}) \subseteq \gtrsim_\pi \\ \{(\mathcal{P}, \mathcal{R})\} & \text{otherwise} \end{cases}$$

Theorem

reduction pair processor with usable rules and argument filters is sound and complete

Example

rewrite rules

$$\begin{aligned} \mathsf{rev}(\mathsf{nil}) &\to \mathsf{nil} & \mathsf{rev}(x : l) &\to \mathsf{rev}_1(x, l) : \mathsf{rev}_2(x, l) \\ \mathsf{rev}_1(x, \mathsf{nil}) &\to x & \mathsf{rev}_1(x, y : l) &\to \mathsf{rev}_1(y, l) \\ \mathsf{rev}_2(x, \mathsf{nil}) &\to \mathsf{nil} & \mathsf{rev}_2(x, y : l) &\to \mathsf{rev}(x : \mathsf{rev}(\mathsf{rev}_2(y, l))) \end{aligned}$$

dependency pairs

$$\begin{aligned} \mathsf{rev}^\sharp(x : l) &\to \mathsf{rev}_1^\sharp(x, l) & \mathsf{rev}_2^\sharp(x, y : l) &\to \mathsf{rev}^\sharp(x : \mathsf{rev}(\mathsf{rev}_2(y, l))) \\ \mathsf{rev}^\sharp(x : l) &\to \mathsf{rev}_2^\sharp(x, l) & \mathsf{rev}_2^\sharp(x, y : l) &\to \mathsf{rev}^\sharp(\mathsf{rev}_2(y, l)) \\ \mathsf{rev}_1^\sharp(x, y : l) &\to \mathsf{rev}_1^\sharp(y, l) & \mathsf{rev}_2^\sharp(x, y : l) &\to \mathsf{rev}_2^\sharp(y, l) \end{aligned}$$

argument filter

$$\pi(:) = [2] \quad \pi(\mathsf{rev}^\sharp) = \pi(\mathsf{rev}) = 1 \quad \pi(\mathsf{rev_1}^\sharp) = \pi(\mathsf{rev_2}^\sharp) = \pi(\mathsf{rev}_2) = 2$$

Outline

Termination of Term Rewriting using Dependency Pairs
Thomas Arts and Jürgen Giesl
TCS 236(1,2), pp. 133 – 178, 2000

Automating the Dependency Pair Method
Nao Hirokawa and Aart Middeldorp
I&C 199(1,2), pp. 172 – 199, 2006

Tyrolean Termination Tool: Techniques and Features
Nao Hirokawa and Aart Middeldorp
I&C 205(4), pp. 474 – 511, 2007

The Dependency Pair Framework: Combining Techniques for Automated Termination Proofs
Jürgen Giesl, René Thiemann and Peter Schneider-Kamp
Proc. 11th LPAR, LNCS 3452, pp. 301 – 331, 2005

Mechanizing and Improving Dependency Pairs
Jürgen Giesl, René Thiemann, Peter Schneider-Kamp and Stephan Falke
JAR 37(3), pp. 155 – 203, 2006

Definition

- $\mathsf{tcap}_{\mathcal{R}}(t) = f(\mathsf{tcap}_{\mathcal{R}}(t_1), \ldots, \mathsf{tcap}_{\mathcal{R}}(t_n))$

  if $t = f(t_1, \ldots, t_n)$ and $f(\mathsf{tcap}_{\mathcal{R}}(t_1), \ldots, \mathsf{tcap}_{\mathcal{R}}(t_n))$ does not unify with any left-hand side of $\mathcal{R}$
- $\mathsf{tcap}_{\mathcal{R}}(t)$ is fresh variable

  otherwise

Definition (Modern Approximation)

$$s \to t \xrightarrow{\mathsf{m}} u \to v \quad \Longleftrightarrow \quad \begin{cases} \mathsf{tcap}_{\mathcal{R}}(t) \text{ and } u \text{ are unifiable} \\ t \text{ and } \mathsf{tcap}_{\mathcal{R}^{-1}}(u) \text{ are unifiable} \end{cases}$$

return