

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

$$\begin{array}{ll}
 f(0) \rightarrow 0 & s(s(0)) \rightarrow f(0) \\
 f(s(0)) \rightarrow s(0) & s(s(s(0))) \rightarrow f(s(0)) \\
 f(s(s(0))) \rightarrow s(s(s(s(s(0)))))) & s(s(s(s(s(s(s(0))))))) \rightarrow f(s(s(0)))
 \end{array}$$

polynomially terminating over \mathbb{N} ?

Solutions

- $f_{\mathbb{N}}(x) = 2x^3 + 2x^2 + x + 1$ $s_{\mathbb{N}}(x) = 2x + 1$ $0_{\mathbb{N}} = 0$
- $f_{\mathbb{N}}(x) = 2x^2 - x + 1$ $s_{\mathbb{N}}(x) = x + 1$ $0_{\mathbb{N}} = 0$

Outline

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

Lemma

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

all proper subterms are terminating

Notation

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

Lemma

$\forall t \in \mathcal{T}_\infty \exists l \rightarrow r \in \mathcal{R} \exists \sigma \exists$ *non-variable* subterm u of r
 $t \xrightarrow{\geq \epsilon}^* l\sigma \xrightarrow{\epsilon} r\sigma \supseteq u\sigma \in \mathcal{T}_\infty$

Corollary

every term in \mathcal{T}_∞ has *defined* root symbol

Lemma

$\forall t \in \mathcal{T}_\infty \quad \exists l \rightarrow r \in \mathcal{R} \quad \exists \sigma \quad \exists \text{ non-variable subterm } u \text{ of } r \text{ with } u \not\triangleleft l$
 $t \xrightarrow{>\epsilon,*} l\sigma \xrightarrow{\epsilon} r\sigma \supseteq u\sigma \in \mathcal{T}_\infty$

(Tentative) Definition

$\mathcal{S} = \{l \rightarrow u \mid l \rightarrow r \in \mathcal{R}, u \triangleleft r \text{ with } \text{root}(u) \text{ defined, } u \not\triangleleft l\}$

Lemma

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

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}

- $\mathcal{F}^\# = \mathcal{F} \cup \{f^\# \mid f \text{ is defined symbol of } \mathcal{R}\}$
- if $t = f(t_1, \dots, t_n)$ with f defined then $t^\# = f^\#(t_1, \dots, t_n)$
- $\mathcal{T}_\infty^\# = \{t^\# \mid t \in \mathcal{T}_\infty\}$ dependency pair
- $\text{DP}(\mathcal{R}) = \{l^\# \rightarrow u^\# \mid l \rightarrow r \in \mathcal{R}, u \sqsubseteq r \text{ with } \text{root}(u) \text{ defined, } u \not\prec l\}$

Lemma

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

Corollary

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

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

Example

rewrite rules

$$0 + y \rightarrow y$$

$$0 \times y \rightarrow 0$$

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

$$s(x) \times y \rightarrow (x \times y) + y$$

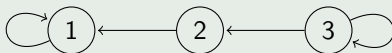
dependency pairs

$$1: \quad s(x) +^{\#} y \rightarrow x +^{\#} y$$

$$2: \quad s(x) \times^{\#} y \rightarrow (x \times y) +^{\#} y$$

$$3: \quad s(x) \times^{\#} y \rightarrow x \times^{\#} y$$

dependency graph



strongly connected components

{1} {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*

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

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

Observation

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

Definition

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

- nodes: dependency pairs of \mathcal{R}
- arrows: $s \rightarrow t \longrightarrow u \rightarrow v$ 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

$$s \rightarrow t \xrightarrow{t} u \rightarrow v \quad \iff \quad \text{root}(t) = \text{root}(u)$$

Better Approximations

are based on

- unification [▶ details](#)
- tree automata techniques

Example

rewrite rules

$$0 + y \rightarrow y$$

$$0 \times y \rightarrow 0$$

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

$$s(x) \times y \rightarrow (x \times y) + y$$

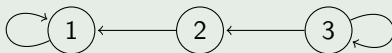
dependency pairs

$$1: \quad s(x) +^{\#} y \rightarrow x +^{\#} y$$

$$2: \quad s(x) \times^{\#} y \rightarrow (x \times y) +^{\#} y$$

$$3: \quad s(x) \times^{\#} y \rightarrow x \times^{\#} y$$

dependency graph



strongly connected components

{1} {3}

Outline

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

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

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

Definition

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

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

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

Theorem

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

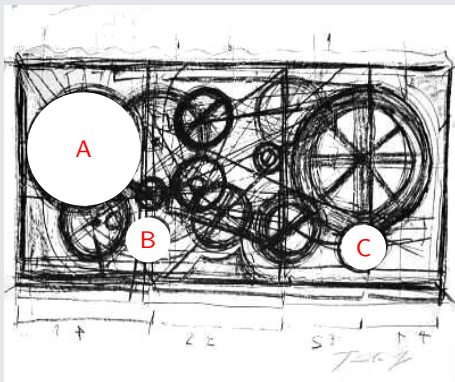
Definition

- **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 $DG(\mathcal{P}, \mathcal{R})$ of DP problem $(\mathcal{P}, \mathcal{R})$ consists of

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

Definition (Dependency Graph Processor)

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

Theorem

dependency graph processor is sound and complete

Outline

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

Definition

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

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

such that all terms t_1, t_2, \dots 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

$$\pi(t_1) \xrightarrow{\mathcal{R}}^* \pi(t_2) \quad ? \quad \pi(t_3) \xrightarrow{\mathcal{R}}^* \pi(t_4) \quad ? \quad \dots$$

Observation

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

Definition

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

Definition (Subterm Criterion Processor)

simple projection π

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

Theorem

subterm criterion processor is sound and complete

Example

rewrite rules

$$\text{ack}(0, y) \rightarrow s(y)$$

$$\text{ack}(s(x), 0) \rightarrow \text{ack}(x, s(0))$$

$$\text{ack}(s(x), s(y)) \rightarrow \text{ack}(x, \text{ack}(s(x), y))$$

dependency pairs

$$\text{ack}^\sharp(s(x), 0) \rightarrow \text{ack}^\sharp(x, s(0))$$

$$\text{ack}^\sharp(s(x), s(y)) \rightarrow \text{ack}^\sharp(x, \text{ack}(s(x), y))$$

$$\text{ack}^\sharp(s(x), s(y)) \rightarrow \text{ack}^\sharp(s(x), y)$$

simple projection

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

Example

rewrite rules

$$\text{ack}(0, y) \rightarrow s(y)$$

$$\text{ack}(s(x), 0) \rightarrow \text{ack}(x, s(0))$$

$$\text{ack}(s(x), s(y)) \rightarrow \text{ack}(x, \text{ack}(s(x), y))$$

dependency pairs

$$\text{ack}^\#(s(x), s(y)) \rightarrow \text{ack}^\#(s(x), y)$$

simple projection

$$\pi(\text{ack}^\#) = 2$$

Outline

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

Definition

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

$$t_1 \succsim t_2 > t_3 \succsim t_4 > \dots$$

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

Question

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

Answers

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

Definition

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

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

Definition (Reduction Pair Processor)

reduction pair $(>, \succsim)$

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

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:

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

for all $a_1, \dots, a_n, b \in A$ and $i \in \{1, \dots, n\}$ 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 $\geq_{\mathcal{A}}$ on terms: $s \geq_{\mathcal{A}} t$ if $[\alpha]_{\mathcal{A}}(s) \geq [\alpha]_{\mathcal{A}}(t)$ for all assignments α

Lemma

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

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

Lemma

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

Example

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

Example

rewrite rules

$$\text{while}(\text{true}, x, y) \rightarrow \text{while}(x > y, p(x), y)$$

$$p(0) \rightarrow 0$$

$$p(s(x)) \rightarrow x$$

$$0 > y \rightarrow \text{false}$$

$$s(x) > 0 \rightarrow \text{true}$$

$$s(x) > s(y) \rightarrow x > y$$

dependency pairs

$$\text{while}^{\#}(\text{true}, x, y) \rightarrow \text{while}^{\#}(x > y, p(x), y)$$

$$s(x) >^{\#} s(y) \rightarrow x >^{\#} y$$

$$\text{while}^{\#}(\text{true}, x, y) \rightarrow x >^{\#} y$$

$$\text{while}^{\#}(\text{true}, x, y) \rightarrow p^{\#}(x)$$

interpretations

$$\text{while}_{\mathbb{N}}(x, y, z) = >_{\mathbb{N}}(x, y) = \text{true}_{\mathbb{N}} = \text{false}_{\mathbb{N}} = 0_{\mathbb{N}} = 0$$

$$\text{while}_{\mathbb{N}}^{\#}(x, y, z) = y + 1 \quad s_{\mathbb{N}}(x) = x + 1 \quad >_{\mathbb{N}}^{\#}(x, y) = p_{\mathbb{N}}(x) = p_{\mathbb{N}}^{\#}(x) = x$$

Example

rewrite rules

$$\text{while}(\text{true}, x, y) \rightarrow \text{while}(x > y, p(x), y)$$

$$p(0) \rightarrow 0$$

$$p(s(x)) \rightarrow x$$

$$0 > y \rightarrow \text{false}$$

$$s(x) > 0 \rightarrow \text{true}$$

$$s(x) > s(y) \rightarrow x > y$$

dependency pairs

$$\text{while}^\#(\text{true}, x, y) \rightarrow \text{while}^\#(x > y, p(x), y)$$

interpretations

$$\text{while}_{\mathbb{Q}}(x, y, z) = \text{false}_{\mathbb{Q}} = 0 \quad >_{\mathbb{Q}}(x, y) = x + \frac{1}{2} \quad \text{true}_{\mathbb{Q}} = \frac{7}{2} \quad 0_{\mathbb{Q}} = 1$$

$$\text{while}^\#_{\mathbb{Q}}(x, y, z) = x + \frac{5}{2}y \quad s_{\mathbb{Q}}(x) = 2x + \frac{7}{2} \quad p_{\mathbb{Q}}(x) = \frac{1}{2}x + 1$$

Definition

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

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

Lemma

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

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

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

Definition

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

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

Lemma

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

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

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

Example

rewrite rules

$$\text{while}(\text{true}, x, y) \rightarrow \text{while}(x > y, p(x), y)$$

$$p(0) \rightarrow 0$$

$$p(s(x)) \rightarrow x$$

$$0 > y \rightarrow \text{false}$$

$$s(x) > 0 \rightarrow \text{true}$$

$$s(x) > s(y) \rightarrow x > y$$

dependency graph analysis

$$\text{while}^\sharp(\text{true}, x, y) \rightarrow \text{while}^\sharp(x > y, p(x), y)$$

$$\text{while}^\sharp(\text{true}, x, y) \rightarrow x >^\sharp y$$

$$s(x) >^\sharp s(y) \rightarrow x >^\sharp y$$

$$\text{while}^\sharp(\text{true}, x, y) \rightarrow p^\sharp(x)$$

subterm criterion

Example

rewrite rules

$$\text{while}(\text{true}, x, y) \rightarrow \text{while}(x > y, p(x), y)$$

$$p(0) \rightarrow 0$$

$$p(s(x)) \rightarrow x$$

$$0 > y \rightarrow \text{false}$$

$$s(x) > 0 \rightarrow \text{true}$$

$$s(x) > s(y) \rightarrow x > y$$

dependency graph analysis

$$\text{while}^\sharp(\text{true}, x, y) \rightarrow \text{while}^\sharp(x > y, p(x), y)$$

arctic interpretation

$$\text{while}_{\mathcal{M}}^\sharp(x, y, z) = x \oplus y \oplus (0)$$

$$s_{\mathbb{N}}(x) = (2) \otimes x \oplus (3)$$

$$\text{while}_{\mathcal{M}}(x, y, z) = x \oplus (2) \otimes y \oplus (0)$$

$$p_{\mathbb{N}}(x) = (-1) \otimes x \oplus (0)$$

$$>_{\mathcal{M}}(x, y) = (-1) \otimes x \oplus (0) \quad \text{true}_{\mathcal{M}} = (2) \quad \text{false}_{\mathcal{M}} = 0_{\mathcal{M}} = (0)$$

Outline

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

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

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

- 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

$$0 - y \rightarrow 0$$

$$x - 0 \rightarrow x$$

$$s(x) - s(y) \rightarrow x - y$$

$$0 \div s(y) \rightarrow 0$$

$$s(x) \div s(y) \rightarrow s((x - y) \div s(y))$$

$$s(x) \div^{\#} s(y) \rightarrow (x - y) \div^{\#} s(y)$$

argument filter $\pi(-) = 1$

LPO with precedence $\div > s$

Outline

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

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 $c(x, y) \gtrsim x, y$ 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 $US(t) = \emptyset$
- if $t = f(t_1, \dots, t_n)$ then $US(t)$ is least set such that
 - 1 $f \in US(t)$
 - 2 $US(t_1) \cup \dots \cup US(t_n) \subseteq US(t)$
 - 3 if $l \rightarrow r \in \mathcal{R}$ and $\text{root}(l) \in US(t)$ then $\mathcal{F}\text{un}(r) \subseteq US(t)$
- usable symbols:

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

- usable rules:

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

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 $c(x, y) \gtrsim x, y$ for fresh symbol c

Theorem

reduction pair processor with usable rules is sound and complete

Puzzle

$\mathcal{C}_{\mathcal{E}}$ -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 $\mathcal{US}_\pi(t) = \emptyset$
- if $t = f(t_1, \dots, t_n)$ 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 $l \rightarrow r \in \mathcal{R}$ and $\text{root}(l) \in \mathcal{US}_\pi(t)$ then $\mathcal{F}\text{un}(r) \subseteq \mathcal{US}_\pi(t)$
- usable symbols with respect to argument filter π :

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

- usable rules with respect to argument filter π :

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

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

$$\begin{array}{ll}
 \text{rev}(\text{nil}) \rightarrow \text{nil} & \text{rev}(x : l) \rightarrow \text{rev}_1(x, l) : \text{rev}_2(x, l) \\
 \text{rev}_1(x, \text{nil}) \rightarrow x & \text{rev}_1(x, y : l) \rightarrow \text{rev}_1(y, l) \\
 \text{rev}_2(x, \text{nil}) \rightarrow \text{nil} & \text{rev}_2(x, y : l) \rightarrow \text{rev}(x : \text{rev}(\text{rev}_2(y, l)))
 \end{array}$$

dependency pairs

$$\begin{array}{ll}
 \text{rev}^\#(x : l) \rightarrow \text{rev}_1^\#(x, l) & \text{rev}_2^\#(x, y : l) \rightarrow \text{rev}^\#(x : \text{rev}(\text{rev}_2(y, l))) \\
 \text{rev}^\#(x : l) \rightarrow \text{rev}_2^\#(x, l) & \text{rev}_2^\#(x, y : l) \rightarrow \text{rev}^\#(\text{rev}_2(y, l)) \\
 \text{rev}_1^\#(x, y : l) \rightarrow \text{rev}_1^\#(y, l) & \text{rev}_2^\#(x, y : l) \rightarrow \text{rev}_2^\#(y, l)
 \end{array}$$

argument filter

$$\pi(\cdot) = [2] \quad \pi(\text{rev}^\#) = \pi(\text{rev}) = 1 \quad \pi(\text{rev}_1^\#) = \pi(\text{rev}_2^\#) = \pi(\text{rev}_2) = 2$$

Outline

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



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

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

Definition (Modern Approximation)

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

[◀ return](#)