



Introduction to Term Rewriting

lecture 10

Aart Middeldorp and Femke van Raamsdonk

Institute of Computer Science
University of Innsbruck

Department of Computer Science
VU Amsterdam



Sunday

introduction, examples, abstract rewriting, equational reasoning, term rewriting

Monday

termination, completion

Tuesday

completion, termination

Wednesday

confluence, modularity, strategies

Thursday

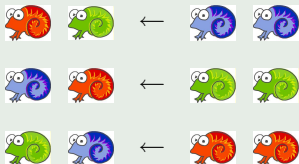
exam, **advanced topics**

Outline

- Matrix Interpretations
- Dependency Pairs
- Semantic Labeling
- Derivational Complexity
- Further Reading

Example

SRS



is terminating ! (RTA open problem # 104)

Definition

algebra \mathcal{M} with well-founded order $>$

- carrier of \mathcal{M} is \mathbb{N}^d with $d > 0$
- interpretations (for every n -ary f)

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

with

- matrices $M_1, \dots, M_n \in \mathbb{N}^{d \times d}$ with $(M_i)_{1,1} \geq 1$ for all $1 \leq i \leq n$
- vector $\vec{f} \in \mathbb{N}^d$
- $(x_1, \dots, x_d)^{\top} > (y_1, \dots, y_d)^{\top} \iff x_1 > y_1 \wedge \bigwedge_{i=2}^d x_i \geq y_i$

Lemma

$(\mathcal{M}, >)$ is well-founded monotone algebra

Example (1)

rewrite rules

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

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

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

matrix interpretation

$$a_{\mathcal{M}}(\vec{x}) = \begin{pmatrix} \mathbf{1} & 0 & 0 & 3 \\ 0 & 0 & 2 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \vec{x} + \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

$$b_{\mathcal{M}}(\vec{x}) = \begin{pmatrix} \mathbf{1} & 2 & 0 & 0 \\ 0 & 2 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \vec{x} + \begin{pmatrix} 0 \\ 2 \\ 0 \\ 0 \end{pmatrix}$$

$$\forall \vec{x} \quad \begin{aligned} a_{\mathcal{M}}(a_{\mathcal{M}}(\vec{x})) &> b_{\mathcal{M}}(c_{\mathcal{M}}(\vec{x})) \\ b_{\mathcal{M}}(b_{\mathcal{M}}(\vec{x})) &> a_{\mathcal{M}}(c_{\mathcal{M}}(\vec{x})) \\ c_{\mathcal{M}}(c_{\mathcal{M}}(\vec{x})) &> a_{\mathcal{M}}(b_{\mathcal{M}}(\vec{x})) \end{aligned}$$

$$c_{\mathcal{M}}(\vec{x}) = \begin{pmatrix} \mathbf{1} & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 2 & 0 & 0 \end{pmatrix} \vec{x} + \begin{pmatrix} 1 \\ 0 \\ 3 \\ 0 \end{pmatrix}$$

Example (2)

rewrite rules

$$f(a) \rightarrow f(b) \quad g(b) \rightarrow g(c) \quad h(c) \rightarrow h(a)$$

matrix interpretation

$$f_{\mathcal{M}}(\vec{x}) = \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix} \vec{x}$$

$$a_{\mathcal{M}} = \begin{pmatrix} 0 \\ 2 \end{pmatrix}$$

$$g_{\mathcal{M}}(\vec{x}) = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \vec{x}$$

$$b_{\mathcal{M}} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$h_{\mathcal{M}}(\vec{x}) = \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix} \vec{x}$$

$$c_{\mathcal{M}} = \begin{pmatrix} 0 \\ 3 \end{pmatrix}$$

Example (3)

rewrite rules

$$f(f(x)) \rightarrow g(x)$$

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

$$f(f(f(f(x)))) \rightarrow i(x, x, x)$$

$$i(a_1, x, b_1) \rightarrow i(x, a_2, b_1)$$

$$i(x, a_3, b_1) \rightarrow i(a_4, x, b_1)$$

$$i(x, x, b_1) \rightarrow i(g(x), b_2, b_2)$$

$$i(g(x), b_3, b_3) \rightarrow i(x, x, b_4)$$

matrix interpretation

$$f_{\mathcal{A}}(\vec{x}) = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \vec{x} + \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$a_{1\mathcal{A}} = a_{3\mathcal{A}} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$b_{1\mathcal{A}} = \begin{pmatrix} 13 \\ 0 \end{pmatrix}$$

$$g_{\mathcal{A}}(\vec{x}) = \begin{pmatrix} 2 & 0 \\ 2 & 0 \end{pmatrix} \vec{x}$$

$$a_{2\mathcal{A}} = a_{4\mathcal{A}} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$b_{2\mathcal{A}} = \begin{pmatrix} 4 \\ 0 \end{pmatrix}$$

$$h_{\mathcal{A}}(\vec{x}) = \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix} \vec{x} + \begin{pmatrix} 3 \\ 1 \end{pmatrix}$$

$$b_{3\mathcal{A}} = \begin{pmatrix} 4 \\ 1 \end{pmatrix}$$

$$b_{4\mathcal{A}} = \begin{pmatrix} 0 \\ 6 \end{pmatrix}$$

$$i_{\mathcal{A}}(\vec{x}, \vec{y}, \vec{z}) = \begin{pmatrix} 2 & 0 \\ 0 & 0 \end{pmatrix} \vec{x} + \begin{pmatrix} 2 & 0 \\ 0 & 0 \end{pmatrix} \vec{y} + \begin{pmatrix} 1 & 2 \\ 1 & 0 \end{pmatrix} \vec{z} + \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Outline

- Matrix Interpretations
- **Dependency Pairs**
 - Processors
 - Dependency Graph
 - Reduction Pairs
 - Argument Filters
 - Subterm Criterion
- Semantic Labeling
- Derivational Complexity
- Further Reading

Definition

DP problem is pair of finite 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

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

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

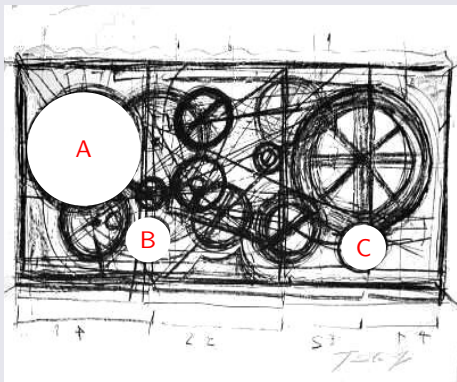
Definitions

- **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

Outline

- Matrix Interpretations
- **Dependency Pairs**
 - Processors
 - **Dependency Graph**
 - Reduction Pairs
 - Argument Filters
 - Subterm Criterion
- Semantic Labeling
 - Model Version
 - Quasi-Model Version
 - Root-Labeling
- Derivational Complexity
 - Polynomial Interpretations
 - Triangular Matrix Interpretations
- Further Reading

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

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
- tree automata techniques

Outline

- Matrix Interpretations
- **Dependency Pairs**
 - Processors
 - Dependency Graph
 - **Reduction Pairs**
 - Argument Filters
 - Subterm Criterion
- Semantic Labeling
 - Model Version
 - Quasi-Model Version
 - Root-Labeling
- Derivational Complexity
 - Polynomial Interpretations
 - Triangular Matrix Interpretations
- 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

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

Example

rewrite rules

$$0 + y \geq_{\mathbb{N}} y$$

$$s(x) + y >_{\mathbb{N}} s(x + y)$$

dependency pairs

$$s(x) +^{\#} y >_{\mathbb{N}} x +^{\#} y$$

polynomial interpretations

$$0_{\mathbb{N}} = 0 \quad s_{\mathbb{N}}(x) = x + 1 \quad +_{\mathbb{N}}(x, y) = +^{\#}_{\mathbb{N}}(x, y) = 2x + y$$

reduction pair processor vs rule removal processor

Outline

- Matrix Interpretations
- **Dependency Pairs**
 - Processors
 - Dependency Graph
 - Reduction Pairs
 - **Argument Filters**
 - Subterm Criterion
- Semantic Labeling
 - Model Version
 - Quasi-Model Version
 - Root-Labeling
- Derivational Complexity
 - Polynomial Interpretations
 - Triangular Matrix Interpretations
- 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

- $s >_{\pi} t \iff \pi(s) > \pi(t)$
- $s \succsim_{\pi} t \iff \pi(s) \succsim \pi(t)$

Lemma

$(>_{\pi}, \succsim_{\pi})$ is reduction pair for every reduction pair $(>, \succsim)$ 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

- Matrix Interpretations
- **Dependency Pairs**
 - Processors
 - Dependency Graph
 - Reduction Pairs
 - Argument Filters
 - **Subterm Criterion**
- Semantic Labeling
 - Model Version
 - Quasi-Model Version
 - Root-Labeling
- Derivational Complexity
 - Polynomial Interpretations
 - Triangular Matrix Interpretations
- 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

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

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

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

simple projection

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

Example

rewrite rules

$$\text{low}(n, \text{nil}) \rightarrow \text{nil}$$

$$\text{low}(n, m : x) \rightarrow \text{if-low}(m \leq n, n, m : x)$$

$$\text{high}(n, \text{nil}) \rightarrow \text{nil}$$

$$\text{high}(n, m : x) \rightarrow \text{if-high}(m \leq n, n, m : x)$$

$$\text{nil} ++ y \rightarrow y$$

$$(n : x) ++ y \rightarrow n : (x ++ y)$$

$$\text{quicksort}(\text{nil}) \rightarrow \text{nil}$$

$$\text{quicksort}(n : x) \rightarrow \text{quicksort}(\text{low}(n, x)) ++ (n : \text{quicksort}(\text{high}(n, x)))$$

$$\text{if-low}(\text{false}, n, m : x) \rightarrow \text{low}(n, x)$$

$$\text{if-low}(\text{true}, n, m : x) \rightarrow m : \text{low}(n, x)$$

$$\text{if-high}(\text{false}, n, m : x) \rightarrow m : \text{high}(n, x)$$

$$\text{if-high}(\text{true}, n, m : x) \rightarrow \text{high}(n, x)$$

$$0 \leq y \rightarrow \text{true}$$

$$s(x) \leq 0 \rightarrow \text{false}$$

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

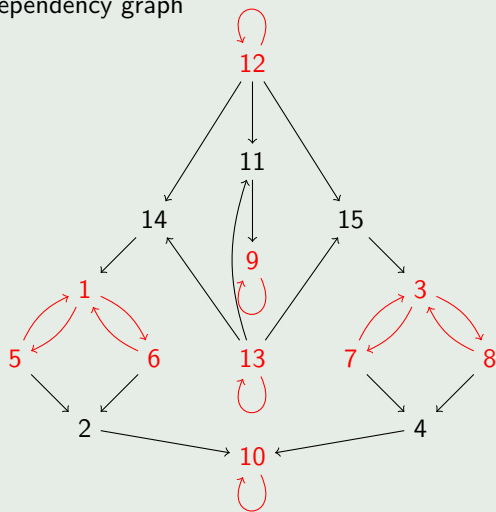
Example (cont'd)

dependency pairs

- | | | | |
|----|--|----|--|
| 1 | $\text{low}^\sharp(n, m : x) \rightarrow \text{if-low}^\sharp(m \leq n, n, m : x)$ | 5 | $\text{if-low}^\sharp(\text{false}, n, m : x) \rightarrow \text{low}^\sharp(n, x)$ |
| 2 | $\text{low}^\sharp(n, m : x) \rightarrow m \leq^\sharp n$ | 6 | $\text{if-low}^\sharp(\text{true}, n, m : x) \rightarrow \text{low}^\sharp(n, x)$ |
| 3 | $\text{high}^\sharp(n, m : x) \rightarrow \text{if-high}^\sharp(m \leq n, n, m : x)$ | 7 | $\text{if-high}^\sharp(\text{false}, n, m : x) \rightarrow \text{high}^\sharp(n, x)$ |
| 4 | $\text{high}^\sharp(n, m : x) \rightarrow m \leq^\sharp n$ | 8 | $\text{if-high}^\sharp(\text{true}, n, m : x) \rightarrow \text{high}^\sharp(n, x)$ |
| 9 | $(n : x) ++^\sharp y \rightarrow x ++^\sharp y$ | 10 | $s(x) \leq^\sharp s(y) \rightarrow x \leq^\sharp y$ |
| 11 | $\text{quicksort}^\sharp(n : x) \rightarrow \text{quicksort}(\text{low}(n, x)) ++^\sharp(n : \text{quicksort}(\text{high}(n, x)))$ | | |
| 12 | $\text{quicksort}^\sharp(n : x) \rightarrow \text{quicksort}^\sharp(\text{low}(n, x))$ | 14 | $\text{quicksort}^\sharp(n : x) \rightarrow \text{low}^\sharp(n, x)$ |
| 13 | $\text{quicksort}^\sharp(n : x) \rightarrow \text{quicksort}^\sharp(\text{high}(n, x))$ | 15 | $\text{quicksort}^\sharp(n : x) \rightarrow \text{high}^\sharp(n, x)$ |

Example (cont'd)

dependency graph



six SCCs

- {9} subterm criterion
- {10} subterm criterion
- {12} AF + LPO
- {13} AF + LPO
- {1, 5, 6} subterm criterion
- {3, 7, 8} subterm criterion

Outline

- Matrix Interpretations
- Dependency Pairs
- **Semantic Labeling**
 - Model Version
 - Quasi-Model Version
 - Root-Labeling
- Derivational Complexity
- Further Reading

Definitions (TRS \mathcal{R} over signature \mathcal{F})

- **semantics**
 - \mathcal{F} -algebra $\mathcal{A} = (A, \{f_{\mathcal{A}}\}_{f \in \mathcal{F}})$
- **labeling**
 - sets of labels $L_f \subseteq A$ for every $f \in \mathcal{F}$
 - labeling functions $\ell_f: A^n \rightarrow L_f$ for every n -ary $f \in \mathcal{F}$ with $L_f \neq \emptyset$
- labeling of terms (for every assignment $\alpha: \mathcal{V} \rightarrow A$)

$$\text{lab}_{\alpha}(t) = \begin{cases} t & \text{if } t \in \mathcal{V} \\ f(\text{lab}_{\alpha}(t_1), \dots, \text{lab}_{\alpha}(t_n)) & \text{if } t = f(t_1, \dots, t_n) \text{ and } L_f = \emptyset \\ f_{\mathbf{a}}(\text{lab}_{\alpha}(t_1), \dots, \text{lab}_{\alpha}(t_n)) & \text{if } t = f(t_1, \dots, t_n) \text{ and } L_f \neq \emptyset \end{cases}$$

with $\mathbf{a} = \ell_f([\alpha]_{\mathcal{A}}(t_1), \dots, [\alpha]_{\mathcal{A}}(t_n))$

- labeled TRS

$$\mathcal{R}_{\text{lab}} = \{ \text{lab}_{\alpha}(\ell) \rightarrow \text{lab}_{\alpha}(r) \mid \ell \rightarrow r \in \mathcal{R} \text{ and } \alpha: \mathcal{V} \rightarrow A \}$$

Example

- TRS \mathcal{R}

$$f(a, b, x) \rightarrow f(x, x, x)$$

- algebra \mathcal{A} is **model** of \mathcal{R}

$$A = \{0, 1\} \quad a_{\mathcal{A}} = 0 \quad b_{\mathcal{A}} = 1 \quad f_{\mathcal{A}}(x, y, z) = 0$$

- labeling ℓ

$$L_a = L_b = \emptyset \quad L_f = A \quad \ell_f(x, y, z) = \begin{cases} 0 & \text{if } x = y \\ 1 & \text{if } x \neq y \end{cases}$$

- TRS \mathcal{R}_{lab}

$$f_1(a, b, x) \rightarrow f_0(x, x, x)$$

is terminating because it lacks dependency pairs

Theorem

TRS \mathcal{R} over signature \mathcal{F} is terminating if

\exists algebra $\mathcal{A} = (A, \{f_{\mathcal{A}}\}_{f \in \mathcal{F}})$

\exists labeling ℓ

such that

1 $\mathcal{R} \subseteq =_{\mathcal{A}}$ “ \mathcal{A} is model of \mathcal{R} ”

2 \mathcal{R}_{lab} is terminating

Theorem (rephrased)

TRS \mathcal{R} , model \mathcal{A} , labeling ℓ (semantic labeling is **sound** and **complete**)

\mathcal{R} is terminating \iff \mathcal{R}_{lab} is terminating

Example

- TRS \mathcal{R}

$$f(a, b, x) \rightarrow f(x, x, x) \quad a \rightarrow c \quad b \rightarrow c \quad f(x, y, z) \rightarrow c$$

- well-founded weakly monotone algebra \mathcal{A}

$$A = \{0, 1, 2\} \quad 2 > 0 \quad 1 > 0 \quad a_{\mathcal{A}} = 1 \quad b_{\mathcal{A}} = 2 \quad c_{\mathcal{A}} = f_{\mathcal{A}}(x, y, z) = 0$$

- weakly monotone labeling ℓ

$$L_a = L_b = L_c = \emptyset \quad L_f = \{0, 1\} \quad \ell_f(x, y, z) = \begin{cases} 1 & \text{if } x = 1 \text{ and } y = 2 \\ 0 & \text{if otherwise} \end{cases}$$

- TRS $\mathcal{R}_{\text{lab}} \cup \mathcal{D}_{\text{ec}}$

$$\begin{array}{l} f_1(a, b, x) \rightarrow f_0(x, x, x) \quad a \rightarrow c \quad b \rightarrow c \quad f_0(x, y, z) \rightarrow c \\ f_1(x, y, z) \rightarrow f_0(x, y, z) \quad f_1(x, y, z) \rightarrow c \end{array}$$

is terminating because dependency graph lacks SCCs

Outline

- Matrix Interpretations
- Dependency Pairs
 - Processors
 - Dependency Graph
 - Reduction Pairs
 - Argument Filters
 - Subterm Criterion
- **Semantic Labeling**
 - Model Version
 - **Quasi-Model Version**
 - Root-Labeling
- Derivational Complexity
 - Polynomial Interpretations
 - Triangular Matrix Interpretations
- Further Reading

Definitions

- well-founded weakly monotone \mathcal{F} -algebra $(\mathcal{A}, >)$ with $\mathcal{A} = (A, \{f_{\mathcal{A}}\}_{f \in \mathcal{F}})$
- weakly monotone labeling functions $\ell_f: A^n \rightarrow L_f$ for every n -ary $f \in \mathcal{F}$ with $L_f \neq \emptyset$
- $\text{Dec} = \{ f_a(x_1, \dots, x_n) \rightarrow f_b(x_1, \dots, x_n) \mid a, b \in L_f \text{ with } a > b \}$

Theorem

TRS \mathcal{R} over signature \mathcal{F} is terminating if

- \exists well-founded weakly monotone algebra $(\mathcal{A}, >, \geq)$
- \exists weakly monotone labeling ℓ

such that

- 1** $\mathcal{R} \subseteq \geq_{\mathcal{A}}$ *" \mathcal{A} is quasi-model of \mathcal{R} "*
- 2** $\mathcal{R}_{lab} \cup \text{Dec}$ is terminating

Example

- TRS \mathcal{R}

$$f(a, b, x) \rightarrow f(x, x, x) \quad a \rightarrow c \quad b \rightarrow c \quad f(x, y, z) \rightarrow c$$

- well-founded weakly monotone algebra \mathcal{A}

$$A = \{0, 1, 2\} \quad 2 > 0 \quad 1 > 0 \quad a_{\mathcal{A}} = 1 \quad b_{\mathcal{A}} = 2 \quad c_{\mathcal{A}} = f_{\mathcal{A}}(x, y, z) = 0$$

- weakly monotone labeling ℓ

$$L_a = L_b = L_c = \emptyset \quad L_f = \{0, 1\} \quad \ell_f(x, y, z) = \begin{cases} 1 & \text{if } x = 1 \text{ and } y = 2 \\ 0 & \text{if otherwise} \end{cases}$$

- TRS $\mathcal{R}_{\text{lab}} \cup \mathcal{D}_{\text{ec}}$

$$\begin{array}{l} f_1(a, b, x) \rightarrow f_0(x, x, x) \quad a \rightarrow c \quad b \rightarrow c \quad f_0(x, y, z) \rightarrow c \\ f_1(x, y, z) \rightarrow f_0(x, y, z) \quad f_1(x, y, z) \rightarrow c \end{array}$$

is terminating because dependency graph lacks SCCs

Theorem (rephrased)

TRS \mathcal{R} , well-founded quasi-model \mathcal{A} , weakly monotone labeling ℓ

$$\mathcal{R} \text{ is terminating} \iff \mathcal{R}_{lab} \cup \mathcal{D}ec \text{ is terminating}$$

Example

- TRS \mathcal{R}

$$\begin{array}{lll} \lambda(x) \circ y \rightarrow \lambda(x \circ (1 \star (y \circ \uparrow))) & id \circ x \rightarrow x & 1 \circ (x \star y) \rightarrow x \\ (x \star y) \circ z \rightarrow (x \circ z) \star (y \circ z) & 1 \circ id \rightarrow 1 & \uparrow \circ (x \star y) \rightarrow y \\ (x \circ y) \circ z \rightarrow x \circ (y \circ z) & \uparrow \circ id \rightarrow \uparrow & \end{array}$$

- well-founded weakly monotone algebra \mathcal{A}
 - carrier $A = \mathbb{N}$ with standard order $>$
 - interpretations

$$\begin{array}{ll} \lambda_{\mathcal{A}}(x) = x + 1 & \circ_{\mathcal{A}}(x, y) = x + y \\ \star_{\mathcal{A}}(x, y) = \max(x, y) & \mathbf{1}_{\mathcal{A}} = \uparrow_{\mathcal{A}} = id_{\mathcal{A}} = 0 \end{array}$$

Example (cont'd)

- weakly monotone labeling ℓ

$$L_\lambda = L_\star = L_1 = L_\uparrow = \emptyset \quad L_\circ = \mathbb{N} \quad \ell_\circ(x, y) = x + y$$

- TRS $\mathcal{R}_{\text{lab}} \cup \mathcal{D}_{\text{ec}} \quad \forall i, j, k \geq 0 \quad \forall l > m \geq 0$

$$\begin{aligned} \lambda(x) \circ_{i+j+1} y &\rightarrow \lambda(x \circ_{i+j} (1 \star (y \circ_j \uparrow))) \\ (x \star y) \circ_{\max(i, j)+k} z &\rightarrow (x \circ_{i+k} z) \star (y \circ_{j+k} z) \\ (x \circ_{i+j} y) \circ_{i+j+k} z &\rightarrow x \circ_{i+j+k} (y \circ_{j+k} z) \\ \text{id} \circ_i x &\rightarrow x & 1 \circ_{\max(i, j)} (x \star y) &\rightarrow x \\ 1 \circ_0 \text{id} &\rightarrow 1 & \uparrow \circ_{\max(i, j)} (x \star y) &\rightarrow y \\ \uparrow \circ_0 \text{id} &\rightarrow \uparrow & x \circ_l y &\rightarrow x \circ_m y \end{aligned}$$

- LPO with precedence

$$\dots > \circ_2 > \circ_1 > \circ_0 > \star, \lambda, 1, \uparrow$$

Extensions

- quasi-model condition required for usable rules only (**predictive** labeling)
- incorporation of argument filters
- DP processors based on semantic labeling

Questions

- how to choose (quasi-)model ?
- how to choose labeling ?

Answers

- ...
- **root-labeling**

Outline

- Matrix Interpretations
- Dependency Pairs
 - Processors
 - Dependency Graph
 - Reduction Pairs
 - Argument Filters
 - Subterm Criterion
- **Semantic Labeling**
 - Model Version
 - Quasi-Model Version
 - **Root-Labeling**
- Derivational Complexity
 - Polynomial Interpretations
 - Triangular Matrix Interpretations
- Further Reading

Definition

- algebra $\mathcal{A}_{\mathcal{F}}$
 - $A_{\mathcal{F}} = \mathcal{F}$
 - $f_{\mathcal{A}_{\mathcal{F}}}(x_1, \dots, x_n) = f \quad \forall n\text{-ary } f \in \mathcal{F}$
- labeling
 - $L_f = \mathcal{F}^n$
 - $\ell_f(x_1, \dots, x_n) = x_1 \cdots x_n$

Example

$$\mathcal{R} \quad a(b(x)) \rightarrow c(x) \qquad \begin{array}{l} a_b(b_a(x)) \rightarrow c_a(x) \\ a_b(b_b(x)) \rightarrow c_b(x) \\ a_b(b_c(x)) \rightarrow c_c(x) \end{array} \quad \mathcal{R}_{\text{rlab}}$$

Problem

no model: $\forall \alpha \quad [\alpha](a(b(x))) = a \neq c = [\alpha](c(x))$

Definitions (TRS \mathcal{R} over signature \mathcal{F})

- **root-preserving** rules

$$\mathcal{R}_p = \{ \ell \rightarrow r \in \mathcal{R} \mid \text{root}(\ell) = \text{root}(r) \}$$

- **root-altering** rules

$$\mathcal{R}_a = \{ \ell \rightarrow r \in \mathcal{R} \mid \text{root}(\ell) \neq \text{root}(r) \}$$

- **flat** contexts

$$\mathcal{FC} = \{ g(x_1, \dots, x_{i-1}, \square, x_i, \dots, x_n) \mid g \in \mathcal{F} \text{ has arity } n \text{ and } 1 \leq i \leq n \}$$

- $\mathcal{FC}(\mathcal{R}) = \mathcal{R}_p \cup \{ C[\ell] \rightarrow C[r] \mid \ell \rightarrow r \in \mathcal{R}_a \text{ and } C \in \mathcal{FC} \}$

Lemma

- \mathcal{R} is terminating if and only if $\mathcal{FC}(\mathcal{R})$ is terminating
- $\mathcal{A}_{\mathcal{F}}$ is model of $\mathcal{FC}(\mathcal{R})$

Example

\mathcal{R}	$a(b(x)) \rightarrow c(x)$	
$\mathcal{FC}(\mathcal{R})$	$a(a(b(x))) \rightarrow a(c(x))$	$a(\square)$
	$b(a(b(x))) \rightarrow b(c(x))$	$b(\square)$
	$c(a(b(x))) \rightarrow c(c(x))$	$c(\square)$
$\mathcal{FC}(\mathcal{R})_{\text{rlab}}$	$a_a(a_b(b_a(x))) \rightarrow a_c(c_a(x))$	
	$a_a(a_b(b_b(x))) \rightarrow a_c(c_b(x))$	
	$a_a(a_b(b_c(x))) \rightarrow a_c(c_c(x))$	
	$b_a(a_b(b_a(x))) \rightarrow b_c(c_a(x))$	
	$b_a(a_b(b_b(x))) \rightarrow b_c(c_b(x))$	
	$b_a(a_b(b_c(x))) \rightarrow b_c(c_c(x))$	
	$c_a(a_b(b_a(x))) \rightarrow c_c(c_a(x))$	
	$c_a(a_b(b_b(x))) \rightarrow c_c(c_b(x))$	
	$c_a(a_b(b_c(x))) \rightarrow c_c(c_c(x))$	

Corollary

\mathcal{R} is terminating if and only if $\mathcal{FC}(\mathcal{R})_{rlab}$ is terminating

Example (1)

- \mathcal{R} $f(a, b, x) \rightarrow f(x, x, x)$
- $\mathcal{FC}(\mathcal{R}) = \mathcal{R}$
- $\mathcal{FC}(\mathcal{R})_{rlab}$ $f_{(a,b,a)}(a, b, x) \rightarrow f_{(a,a,a)}(x, x, x)$
 $f_{(a,b,b)}(a, b, x) \rightarrow f_{(b,b,b)}(x, x, x)$
 $f_{(a,b,f)}(a, b, x) \rightarrow f_{(f,f,f)}(x, x, x)$
- $\mathcal{FC}(\mathcal{R})_{rlab}$ is terminating because it lacks dependency pairs

Example (2)

- $\mathcal{R} \quad f(f(x)) \rightarrow f(g(f(x)))$
- $\mathcal{FC}(\mathcal{R}) = \mathcal{R}$
- $\mathcal{FC}(\mathcal{R})_{\text{rlab}} \quad f_f(f_f(x)) \rightarrow f_g(g_f(f_f(x)))$
 $f_f(f_g(x)) \rightarrow f_g(g_f(f_g(x)))$
- $\mathcal{FC}(\mathcal{R})_{\text{rlab}}$ is polynomially terminating

Outline

- Matrix Interpretations
- Dependency Pairs
- Semantic Labeling
- Derivational Complexity
 - Polynomial Interpretations
 - Triangular Matrix Interpretations
- Further Reading

Definitions

- **derivation length** $dl(t) = \max \{ n \mid \exists u: t \rightarrow^n u \}$
- **derivational complexity** $dc(n) = \max \{ dl(t) \mid |t| = n \}$

Example

rewrite rule $a(b(x)) \rightarrow b(a(x))$

term	derivation length	value
$a(b(b(b(c))))$	3	6
$a(a(a(b(b(b(c))))))$	9	24
$a^m(b^n(c))$	mn	$2^m n$

polynomial interpretation

$$a_{\mathbb{N}}(x) = 2x \quad b_{\mathbb{N}}(x) = x + 1 \quad c_{\mathbb{N}} = 0$$

Theorem

	<i>interpretation in \mathbb{N}</i>	<i>bound on derivational complexity</i>
	<i>polynomial</i>	<i>double-exponential</i>
$a_1x_1 + \dots + a_nx_n + b$	<i>linear</i>	<i>exponential</i>
$x_1 + \dots + x_n + b$	<i>strongly linear</i>	<i>linear</i>

Example

rewrite system

$$\begin{array}{lll}
 x + 0 \rightarrow x & d(0) \rightarrow 0 & q(0) \rightarrow 0 \\
 x + s(y) \rightarrow s(x + y) & d(s(x)) \rightarrow s(s(d(x))) & q(s(x)) \rightarrow q(x) + s(d(x))
 \end{array}$$

interpretations

$$0_{\mathbb{N}} = 2 \quad s_{\mathbb{N}}(x) = x + 1 \quad +_{\mathbb{N}}(x, y) = x + 2y \quad d_{\mathbb{N}}(x) = 3x \quad q_{\mathbb{N}}(x) = x^3$$

Outline

- Matrix Interpretations
- Dependency Pairs
 - Processors
 - Dependency Graph
 - Reduction Pairs
 - Argument Filters
 - Subterm Criterion
- Semantic Labeling
 - Model Version
 - Quasi-Model Version
 - Root-Labeling
- Derivational Complexity
 - Polynomial Interpretations
 - **Triangular Matrix Interpretations**
- Further Reading

Definition

triangular matrix interpretation \mathcal{M} over \mathbb{N}

- carrier of \mathcal{M} is \mathbb{N}^d with $d > 0$
- interpretations (for every n -ary f)

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

with

- matrices $M_1, \dots, M_n \in \mathbb{N}^{d \times d}$ with

$$(M_i)_{1,1} \geq 1 \quad (M_i)_{j,k} = 0 \text{ for all } j > k \quad (M_i)_{j,j} \leq 1 \text{ for all } j$$

for all $1 \leq i \leq n$

- vector $\vec{f} \in \mathbb{N}^d$
- $(x_1, \dots, x_d)^T > (y_1, \dots, y_d)^T \iff x_1 > y_1 \wedge \bigwedge_{i=2}^d x_i \geq y_i$

Theorem

if $\mathcal{R} \subseteq \succ_{\mathcal{M}}$ for triangular matrix interpretation \mathcal{M} of dimension d then

$$\text{dc}(n) = \mathcal{O}(n^d)$$

Example

rewrite rules

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

triangular matrix interpretation

$$a_{\mathcal{M}}(\vec{x}) = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \vec{x} + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \quad c_{\mathcal{M}}(\vec{x}) = \begin{pmatrix} 1 & 1 & 3 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \vec{x} + \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

$$b_{\mathcal{M}}(\vec{x}) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \vec{x} + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

Outline

- Matrix Interpretations
- Dependency Pairs
- Semantic Labeling
- Derivational Complexity
- Further Reading



Matrix Interpretations for Proving Termination of Term Rewriting

Jörg Endrullis, Johannes Waldmann and Hans Zantema

JAR 40(2,3), pp. 195 – 220, 2008



Mechanizing and Improving Dependency Pairs

Jürgen Giesl, René Thiemann, Peter Schneider-Kamp and Stephan Falke

JAR 37(3), pp. 155 – 203, 2006



Tyrolean Termination Tool: Techniques and Features

Nao Hirokawa and Aart Middeldorp

I&C 205(4), pp. 474 – 511, 2007



Termination of Term Rewriting by Semantic Labelling

Hans Zantema

FI 24(1,2), pp. 89 – 105, 1995



Root-Labeling

Christian Sternagel and Aart Middeldorp

Proc. 19th RTA, LNCS 5117, pp. 336 – 350, 2008



Complexity Analysis of Term Rewriting Based on Matrix and Context Dependent Interpretations

Georg Moser, Andreas Schnabl and Johannes Waldmann

Proc. 28th FSTTCS, LIPI 2, pp. 304 – 315, 2008

Termination Tools

- AProVE
- CiME
- Jambox
- Matchbox
- MuTerm
- TTT2
- VMTL
- ...

Complexity Tools

- CaT
- TCT

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)

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