

Advanced Topics in Termination

ISR 2009 – lecture 4

Aart Middeldorp

Institute of Computer Science
University of Innsbruck



Derivational Complexity

Outline

- Derivational Complexity
 - Polynomial Interpretations
 - Context Dependent Interpretations
 - Triangular Matrix Interpretations
- Decidable Classes
- Certification
- Applications
- Further Reading

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/>

Derivational Complexity

Polynomial Interpretations

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

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

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

- Derivational Complexity
 - Polynomial Interpretations
 - Context Dependent Interpretations
 - Triangular Matrix Interpretations
- Decidable Classes
- Certification
- Applications
 - Program Termination
 - Completion
- Further Reading

Definitions

- context dependent \mathcal{F} -algebra (CDA) \mathcal{C} assign to every n -ary $f \in \mathcal{F}$
 - interpretation function $f_{\mathcal{C}}: \mathbb{R}_+ \times \mathbb{R}_0^n \rightarrow \mathbb{R}_0$
 - parameter functions $f_c^i: \mathbb{R}_+ \rightarrow \mathbb{R}_+$ for $1 \leq i \leq n$
- Δ -assignment is mapping $\alpha: \mathbb{R}_+ \times \mathcal{V} \rightarrow \mathbb{R}_0$
- extension to terms

$$[\alpha, \Delta]_{\mathcal{C}}(t) = \begin{cases} \alpha(\Delta, t) & \text{if } t \in \mathcal{V} \\ f_{\mathcal{C}}(\Delta, [\alpha, f_c^1(\Delta)]_{\mathcal{C}}(t_1), \dots, [\alpha, f_c^n(\Delta)]_{\mathcal{C}}(t_n)) & \text{if } t = f(t_1, \dots, t_n) \end{cases}$$

- relation $>_{\mathcal{C}}$ on terms:
 $s >_{\mathcal{C}} t$ if $[\alpha, \Delta]_{\mathcal{C}}(s) >_{\Delta} [\alpha, \Delta]_{\mathcal{C}}(t)$ for all $\Delta \in \mathbb{R}_+$ and Δ -assignments α

Definition

CDA \mathcal{C} is Δ -monotone if

$$f_{\mathcal{C}}(\Delta, a_1, \dots, a_i, \dots, a_n) >_{\Delta} f_{\mathcal{C}}(\Delta, a_1, \dots, b, \dots, a_n)$$

for all $\Delta \in \mathbb{R}_+$, $a_1, \dots, a_n, b \in \mathbb{R}_0$ and $i \in \{1, \dots, n\}$ with $a_i >_{f_c^i(\Delta)} b$

Theorem

if $\mathcal{R} \subseteq >_{\mathcal{C}}$ for Δ -monotone CDA \mathcal{C} then \mathcal{R} is terminating and

$$dl(t) \leq \liminf_{\Delta \in \mathbb{R}_+} \frac{[\alpha, \Delta]_{\mathcal{C}}(t)}{\Delta}$$

for all terms t

Example

rewrite rule $a(b(x)) \rightarrow b(a(x))$ CDA \mathcal{C}

$$a_{\mathcal{C}}[\Delta](x) = (1 + \Delta)x \quad a_{\mathcal{C}}^1(\Delta) = \frac{\Delta}{1 + \Delta}$$

$$b_{\mathcal{C}}[\Delta](x) = x + 1 \quad b_{\mathcal{C}}^1(\Delta) = \Delta$$

derivational complexity

- $\liminf_{\Delta \in \mathbb{R}_+} \frac{[\alpha, \Delta]_{\mathcal{C}}(a^m(b^n(c)))}{\Delta} = \liminf_{\Delta \in \mathbb{R}_+} \left(\frac{1}{\Delta} + n\right)m = nm = dl(a^m(b^n(c)))$
- $[\alpha, \Delta]_{\mathcal{C}}(a^m(b^n(c))) = (1 + \Delta n)m$

Definition

 Δ -linear interpretation is CDA \mathcal{C} with

$$f_{\mathcal{C}}(\Delta, x_1, \dots, x_n) = \sum_{i=1}^n (a_i + b_i \Delta)x_i + c + d\Delta \quad f_{\mathcal{C}}^i(\Delta) = \frac{\Delta}{a_i + b_i \Delta}$$

Theorem

if $\mathcal{R} \subseteq >_{\mathcal{C}}$ for Δ -linear interpretation \mathcal{C} then $dc(n) = 2^{\mathcal{O}(n)}$

Example

rewrite rule $a(b(x)) \rightarrow b(b(a(x)))$ Δ -linear interpretation \mathcal{C}

$$a_{\mathcal{C}}[\Delta](x) = (2 + 2\Delta)x \quad a_{\mathcal{C}}^1(\Delta) = \frac{\Delta}{2 + 2\Delta}$$

$$b_{\mathcal{C}}[\Delta](x) = x + 1 \quad b_{\mathcal{C}}^1(\Delta) = \Delta$$

Definition

 Δ -restricted interpretation is CDA \mathcal{C} with

$$f_{\mathcal{C}}(\Delta, x_1, \dots, x_n) = \sum_{i=1}^n (a_i + b_i \Delta)x_i + c + d\Delta \quad f_{\mathcal{C}}^i(\Delta) = \frac{\Delta}{a_i + b_i \Delta}$$

and $a_i \in \{0, 1\}$

Theorem

if $\mathcal{R} \subseteq >_{\mathcal{C}}$ for Δ -restricted interpretation \mathcal{C} then $dc(n) = \mathcal{O}(n^2)$

Example

rewrite rule $a(a(x)) \rightarrow a(b(a(x)))$ Δ -restricted interpretation \mathcal{C}

$$a_{\mathcal{C}}[\Delta](x) = 2\Delta x + 2 \quad a_{\mathcal{C}}^1(\Delta) = \frac{1}{2} \quad b_{\mathcal{C}}[\Delta](x) = \Delta x \quad b_{\mathcal{C}}^1(\Delta) = 1$$

Outline

- Derivational Complexity
 - Polynomial Interpretations
 - Context Dependent Interpretations
 - Triangular Matrix Interpretations
- Decidable Classes
- Certification
- Applications
 - Program Termination
 - Completion
- 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 $\vec{f} \in \mathbb{N}^d$
- $(x_1, \dots, x_d)^T > (y_1, \dots, y_d)^T \iff x_1 > y_1 \wedge \bigwedge_{i=2}^d x_i \geq y_i$

Outline

- Derivational Complexity
- Decidable Classes
- Certification
- Applications
- Further Reading

Theorem

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

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

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

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

Theorem (Matiyasevich and Sénizergues)

termination is *undecidable* for *three-rule SRSs*

Open Problem

is termination decidable for *one-rule* SRSs ?

Theorem (Dershowitz)

termination is *decidable* for *right-ground TRSs*

Definition

- term is **shallow** if variables appear only at depth 0 or 1
- rewrite rule $\ell \rightarrow r$ is **right-shallow** if r is shallow
- rewrite rule $\ell \rightarrow r$ is **left-shallow** if ℓ is shallow
- rewrite rule $\ell \rightarrow r$ is **shallow** if it is left- and right-shallow

Theorem

- termination is decidable for **left-linear shallow TRSs**
- termination is decidable for **right-linear right-shallow TRSs**
- termination is undecidable for **right-shallow TRSs**

Definition

- term is **shallow** if variables appear only at depth 0 or 1
- rewrite rule $\ell \rightarrow r$ is **right-shallow** if r is shallow
- rewrite rule $\ell \rightarrow r$ is **left-shallow** if ℓ is shallow
- rewrite rule $\ell \rightarrow r$ is **shallow** if it is left- and right-shallow

Theorem

- termination is decidable for **left-linear shallow TRSs**
- termination is decidable for **right-linear right-shallow TRSs**
- termination is undecidable for **right-shallow TRSs**

Theorem

termination is undecidable for right-shallow TRSs

Proof

reduction from PCP

PCP instance $P = \{(u_1, v_1), \dots, (u_n, v_n)\} \subseteq \Gamma^* \times \Gamma^*$

TRS \mathcal{R}_P

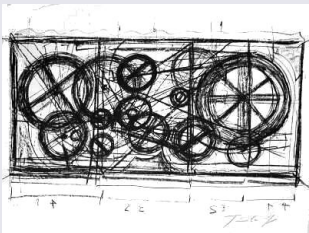
$$\begin{aligned} f(x) &\rightarrow g(x, x, x) \\ g(x, u_i(y), v_i(z)) &\rightarrow h(x, y, z) && \text{for } 1 \leq i \leq n \\ h(x, u_i(y), v_i(z)) &\rightarrow h(x, y, z) && \text{for } 1 \leq i \leq n \\ h(x, c, c) &\rightarrow f(x) \end{aligned}$$

P has solution $\iff \mathcal{R}_P$ is non-terminating

Outline

- Derivational Complexity
- Decidable Classes
- Certification
- Applications
- Further Reading

Termination Tools



- termination tools are **complex**
- termination tools contain **bugs**
- termination tool output is **unreadable**
- termination tool output must be **certified**

Certification of Termination

- Coccinelle/CiME
- CoLoR/Rainbow
- IsaFoR/CeTA
- PVS

Coccinelle/CiME

- Coccinelle: formalization of rewriting in Coq
- shallow embedding
- CiME: generates separate Coq proof for each TRS

CoLoR/Rainbow

- CoLoR: formalization of rewriting in Coq
- deep embedding
- Rainbow: generates separate Coq proof for each TRS

Problems

- calling proof assistant for every proof is **slow**
- error messages **not readable** since generated inside Coq
- goal is generated by tool (correct proof, but for wrong TRS)

IsaFoR/CeTA

- Isabelle Formalization of Rewriting
- Certified Termination Analysis
 - input:* TRS and proof (both in XML format)
 - output:* certified yes (if proof is correct), no (if proof is incorrect)
- CeTA is generated from IsaFoR, using Isabelle's **code-generation** facilities
- certification is **fast**
- error messages use terminology from rewriting

Example

$$\text{add}(0, y) \rightarrow y$$

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

Coccinelle/CiME (Ad Hoc Data Type)

```

Rule_0
  (Var 1)
  (Term id_add [Term id_0 []; Var 1])

Rule_1
  (Term id_s [Term id_add [Var 0; Var 1]])
  (Term id_add [Term id_s [Var 0]; Var 1])

```

Example

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

CoLoR/Rainbow (List of Rules)

```
[Rule (add 0 (V 1)) (V 1);
 Rule (add (s (V 0)) (V 1)) (s (add (V 0) (V 1)))]
```

IsaFoR/CeTA (Set of Pairs of Terms)

```
{(Fun "add" [Fun "0" [], Var "y"], Var "y"),
 (Fun "add" [Fun "s" [Var "x"], Var "y"],
  Fun "s" [Fun "add" [Var "x", Var "y"]])}
```

Observation

programs can be translated to TRSs, DP problems, TRSs with built-in integers, ...

Logic Programming

- pure logic programs
- Prolog programs

Functional Programming

- Haskell programs with starting expression

Imperative Programming

- Java bytecode

Further Information

- AProVE

Outline

- Derivational Complexity
- Decidable Classes
- Certification
- Applications
 - Program Termination
 - Completion
- Further Reading

Outline

- Derivational Complexity
 - Polynomial Interpretations
 - Context Dependent Interpretations
 - Triangular Matrix Interpretations
- Decidable Classes
- Certification
- Applications
 - Program Termination
 - Completion
- Further Reading

Definition (Standard Completion)

set \mathcal{E} of equations set \mathcal{R} of rewrite rules **reduction order $>$**
 inference system **SC** consists of six rules

delete	$\frac{\mathcal{E} \cup \{s \approx s\}, \mathcal{R}}{\mathcal{E}, \mathcal{R}}$	
compose	$\frac{\mathcal{E}, \mathcal{R} \cup \{s \rightarrow t\}}{\mathcal{E}, \mathcal{R} \cup \{s \rightarrow u\}}$	if $t \rightarrow_{\mathcal{R}} u$
simplify	$\frac{\mathcal{E} \cup \{s \approx t\}, \mathcal{R}}{\mathcal{E} \cup \{s \approx u\}, \mathcal{R}}$	if $t \rightarrow_{\mathcal{R}} u$
orient	$\frac{\mathcal{E} \cup \{s \approx t\}, \mathcal{R}}{\mathcal{E}, \mathcal{R} \cup \{s \rightarrow t\}}$	if $s > t$
collapse	$\frac{\mathcal{E}, \mathcal{R} \cup \{t \rightarrow s\}}{\mathcal{E} \cup \{u \approx s\}, \mathcal{R}}$	if $t \rightarrow_{\mathcal{R}} u$ using $l \rightarrow r \in \mathcal{R}$ with $t \triangleright l$
deduce	$\frac{\mathcal{E}, \mathcal{R}}{\mathcal{E} \cup \{s \approx t\}, \mathcal{R}}$	if $s \leftarrow_{\mathcal{R}} u \rightarrow_{\mathcal{R}} t$

Definitions

completion procedure is program that takes as input set of equations \mathcal{E} and reduction order $>$ and generates (finite or infinite) **run**

$$(\mathcal{E}_0, \mathcal{R}_0) \vdash_{SC} (\mathcal{E}_1, \mathcal{R}_1) \vdash_{SC} (\mathcal{E}_2, \mathcal{R}_2) \vdash_{SC} \dots$$

with $\mathcal{E}_0 = \mathcal{E}$ and $\mathcal{R}_0 = \emptyset$

- \mathcal{E}_ω is set of **persistent** equations: $\mathcal{E}_\omega = \bigcup_{i \geq 0} \bigcap_{j \geq i} \mathcal{E}_j$
- \mathcal{R}_ω is set of persistent rules
- run **succeeds** if $\mathcal{E}_\omega = \emptyset$ and \mathcal{R}_ω is confluent and terminating
- run **fails** if $\mathcal{E}_\omega \neq \emptyset$

Definition

completion procedure is **correct** if every run that does not fail succeeds

Definitions

- run $(\mathcal{E}_0, \mathcal{R}_0) \vdash_{SC} (\mathcal{E}_1, \mathcal{R}_1) \vdash_{SC} (\mathcal{E}_2, \mathcal{R}_2) \vdash_{SC} \dots$ is **fair** if

$$CP(\mathcal{R}_\omega) \subseteq \bigcup_{i \geq 0} \mathcal{E}_i$$

- completion procedure is fair if every run that does not fail is fair

Theorem

every fair completion procedure is correct

Remark

reduction order is used to determine orientation of rewrite rules

Definition (Standard Completion with Termination Tools)

inference system **SCtt** consists of six rules

delete	$\frac{\mathcal{E} \cup \{s \approx s\}, \mathcal{R}, \mathcal{C}}{\mathcal{E}, \mathcal{R}, \mathcal{C}}$	compose	$\frac{\mathcal{E}, \mathcal{R} \cup \{s \rightarrow t\}, \mathcal{C}}{\mathcal{E}, \mathcal{R} \cup \{s \rightarrow u\}, \mathcal{C}}$	if $t \rightarrow_{\mathcal{R}} u$
simplify	$\frac{\mathcal{E} \cup \{s \approx t\}, \mathcal{R}, \mathcal{C}}{\mathcal{E} \cup \{s \approx u\}, \mathcal{R}, \mathcal{C}}$			if $t \rightarrow_{\mathcal{R}} u$
orient	$\frac{\mathcal{E} \cup \{s \approx t\}, \mathcal{R}, \mathcal{C}}{\mathcal{E}, \mathcal{R} \cup \{s \rightarrow t\}, \mathcal{C} \cup \{s \rightarrow t\}}$			if $\mathcal{C} \cup \{s \rightarrow t\}$ is terminating
collapse	$\frac{\mathcal{E}, \mathcal{R} \cup \{t \rightarrow s\}, \mathcal{C}}{\mathcal{E} \cup \{u \approx s\}, \mathcal{R}, \mathcal{C}}$			if $t \rightarrow_{\mathcal{R}} u$ using $l \rightarrow r \in \mathcal{R}$ with $t \triangleright l$
deduce	$\frac{\mathcal{E}, \mathcal{R}, \mathcal{C}}{\mathcal{E} \cup \{s \approx t\}, \mathcal{R}, \mathcal{C}}$			if $s \leftarrow_{\mathcal{R}} u \rightarrow_{\mathcal{R}} t$

Problem

if $\mathcal{E}, \emptyset \vdash_{SCtt}^* \emptyset, \mathcal{R}$ is fair then \mathcal{R} need not be confluent

Theorem

- $\mathcal{E}, \emptyset, \emptyset \vdash_{SCtt}^* \emptyset, \mathcal{R}, \mathcal{C} \implies \mathcal{E}, \emptyset \vdash_{SC}^* \emptyset, \mathcal{R}$ with respect to $> = \rightarrow_C^+$
- $\mathcal{E}, \emptyset, \emptyset \vdash_{SCtt}^* \emptyset, \mathcal{R}, \mathcal{C} \iff \mathcal{E}, \emptyset \vdash_{SC}^* \emptyset, \mathcal{R}$

Remarks

- $SCtt$ is implemented in **Slothrop**
- completion of **CGE₂**



$$\begin{array}{ll} e \cdot x \approx x & f(x \cdot y) \approx f(x) \cdot f(y) \\ x^- \cdot x \approx e & g(x \cdot y) \approx g(x) \cdot g(y) \\ (x \cdot y) \cdot z \approx x \cdot (y \cdot z) & f(x) \cdot g(y) \approx g(y) \cdot f(x) \end{array}$$






is Slothrop's defining achievement

- CGE₂ cannot be completed by state-of-the-art implementations of SC like **Waldmeister**
- **multi-completion** with termination tools is more efficient (**MKBtt**)

Outline

- Derivational Complexity
- Decidable Classes
- Certification
- Applications
- Further Reading

-  [Termination Proofs and the Length of Derivations](#)
Dieter Hofbauer and Clemens Lautemann
Proc. 3rd RTA, LNCS 355, pp. 167 – 177, 1989
-  [Termination Proofs by Context-Dependent Interpretations](#)
Dieter Hofbauer
Proc. 12th RTA, LNCS 2051, pp. 108 – 121, 2001
-  [Proving Quadratic Derivational Complexities Using Context Dependent Interpretations](#)
Georg Moser and Andreas Schnabl
Proc. 18th RTA, LNCS 5117, pp. 276 – 290, 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 of Rewriting with Right-Flat Rules](#)
Guillem Godoy, Eduard Huntingford and Ashish Tiwari
Proc. 18th RTA, LNCS 4533, pp. 200 – 213, 2007

-  [Certification of Termination Proofs using CeTA](#)
René Thiemann and Christian Sternagel
Proc. 22nd TPHOLs, LNCS 5674, 2009, to appear
-  [Automated Termination Proofs for Logic Programs by Term Rewriting](#)
Peter Schneider-Kamp, Jürgen Giesl, Alexander Serebrenik and René Thiemann
ACM ToCL, 2009, to appear
-  [Automated Termination Analysis for Haskell: From Term Rewriting to Programming Languages](#)
Jürgen Giesl, Stephan Swiderski, Peter Schneider-Kamp and René Thiemann
Proc. 17th RTA, LNCS 4098, pp. 297 – 312, 2006
-  [Slothrop: Knuth-Bendix Completion with a Modern Termination Checker](#)
Ian Wehrman, Aaron Stump and Edwin Westbrook
Proc. 17th RTA, LNCS 4098, pp. 287 – 296, 2006
-  [Multi-Completion with Termination Tools \(System Description\)](#)
Haruhiko Sato, Sarah Winkler, Masahito Kurihara and Aart Middeldorp
Proc. 4th IJCAR, LNAI 5195, pp. 306 – 312, 2008

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

- slides (with active hyperlinks)
- papers
- tools
- ISR 2009 version of T_1T_2

Problem

if $\mathcal{E}, \emptyset \vdash_{SCtt}^* \emptyset, \mathcal{R}$ is fair then \mathcal{R} need not be confluent

◀ return

Example

1

$129 \rightarrow 1$	$127 \rightarrow 128$	$34 \rightarrow 54$
$14 \rightarrow 4$	$284 \rightarrow 294$	$26 \rightarrow 27$
$254 \rightarrow 274$		$23 \rightarrow 29$

- LPO with precedence $3 > 5 > 6 > 7 > 8 > 9$
- confluent and terminating

2

$123 \rightarrow 1$	$127 \leftarrow 128$	$34 \rightarrow 54$
$14 \rightarrow 4$	$284 \rightarrow 234$	$26 \rightarrow 27$
$254 \rightarrow 274$		$23 \leftarrow 29$

- LPO with precedence $8 > 9 > 3 > 5 > 6 > 7$
- terminating but not confluent: $1274 \leftarrow 1254 \leftarrow 1234 \rightarrow 14 \rightarrow 4$