Tutorial on Term Rewriting and Narrowing

Aart Middeldorp

University of Tsukuba

http://www.score.is.tsukuba.ac.jp/~ami

OVERVIEW

- examples
- term rewriting
- termination
- confluence
- completion
- strategies
- narrowing
- modularity

- further reading

OVERVIEW

- examples
- term rewriting
- termination
- confluence
- completion
- strategies
- narrowing
- modularity

- further reading

signature $\quad 0$ constant $\quad \mathsf{s}$ unary $\quad + \quad \times$ binary

equations

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

rewriting

$$\begin{aligned} \mathsf{s}(0) + \mathsf{s}(\underbrace{\mathsf{s}(0) \times \mathsf{s}(\mathsf{s}(0))}_{\text{redex}}) &\to \mathsf{s}(0) + \mathsf{s}(\underline{0 \times \mathsf{s}(\mathsf{s}(0))} + \mathsf{s}(\mathsf{s}(0))) \\ &\to \mathsf{s}(0) + \mathsf{s}(\underline{0 + \mathsf{s}(\mathsf{s}(0))}) \\ &\to \underline{\mathsf{s}(0) + \mathsf{s}(\mathsf{s}(\mathsf{s}(0)))} \\ &\to \mathsf{s}(\underline{0 + \mathsf{s}(\mathsf{s}(\mathsf{s}(0)))}) \\ &\to \underbrace{\mathsf{s}(\mathsf{s}(\mathsf{s}(\mathsf{s}(0))))}_{\text{normal form}} \end{aligned}$$

use equations from left to right and matching signature $\quad 0 \;\; \mathsf{fib}$ constants $\quad \mathsf{s}$ unary $\quad + \;\; \mathsf{nth} \;\; \mathsf{f} \;\; :$ binary

rewrite rules

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

rewriting

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

$$\mathsf{nth}(\mathsf{s}(0),\underline{\mathsf{fib}}) \to^{\omega}$$

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

equations

$$\begin{aligned} 0 - y &\approx 0 \\ x - 0 &\approx x \qquad \mathcal{E} \\ \mathsf{s}(x) - \mathsf{s}(y) &\approx x - y \end{aligned}$$

$$\exists\, x\, y \quad \mathsf{s}(\mathsf{s}(x-y)) \approx_{\mathcal{E}} \mathsf{s}(x) \; ?$$

equation solving modulo equational theory $\mathcal{E}$

narrowing $\quad \mathsf{s}(\mathsf{s}(\underline{x-y})) \approx \mathsf{s}(x) \overset{\sigma_1}{\rightarrowtail} \mathsf{s}(\mathsf{s}(x_1 - y_1)) \approx \mathsf{s}(\mathsf{s}(x_1))$

$\sigma_1 : \quad x \mapsto \mathsf{s}(x_1) \quad y \mapsto \mathsf{s}(y_1)$

use equations from left to right and unification $\quad \overset{\sigma_2}{\rightarrowtail} \mathsf{s}(\mathsf{s}(0)) \approx \mathsf{s}(\mathsf{s}(0))$

$\sigma_2 : \quad x_1 \mapsto 0$

solution $\quad \sigma_1\sigma_2\lceil\{x,y\} : \quad x \mapsto \mathsf{s}(0) \quad y \mapsto \mathsf{s}(y_1)$

$$\mathcal{E} \qquad \begin{aligned} \mathsf{e} \cdot x &\approx x \\ x^- \cdot x &\approx \mathsf{e} \\ (x \cdot y) \cdot z &\approx x \cdot (y \cdot z) \end{aligned} \qquad \text{group theory}$$

$$\mathsf{e}^- \approx_{\mathcal{E}} \mathsf{e} \qquad (x \cdot y)^- \approx_{\mathcal{E}} y^- \cdot x^-$$

$$\mathcal{R} \qquad \begin{aligned} \mathsf{e} \cdot x &\to x & x \cdot \mathsf{e} &\to x \\ x^- \cdot x &\to \mathsf{e} & x \cdot x^- &\to \mathsf{e} \\ (x \cdot y) \cdot z &\to x \cdot (y \cdot z) & x^{--} &\to x \\ \mathsf{e}^- &\to \mathsf{e} & (x \cdot y)^- &\to y^- \cdot x^- \\ x^- \cdot (x \cdot y) &\to y & x \cdot (x^- \cdot y) &\to y \end{aligned}$$

① equation $s \approx t$ is valid in $\mathcal{E}$ iff s and t have same $\mathcal{R}$-normal form

② $\mathcal{R}$ admits no infinite computations

① + ② $\implies$ $\mathcal{E}$ has decidable validity problem

QUESTIONS

➜ semantics ?
equational logic

➜ are answers unique ?
confluence

➜ do all computations end in answer ?
termination

➜ how to solve validity problems by rewriting ?
completion

➜ how to compute answers ?
strategies

➜ how to solve equations ?
narrowing

Overview

- examples
- term rewriting
- termination
- confluence
- completion
- strategies
- narrowing
- modularity

- further reading

Term Rewrite Systems

equational system (ES) is pair $(\mathcal{F}, \mathcal{E})$

- $\mathcal{F}$ signature
- $\mathcal{E}$ set of equations between terms in $\mathcal{T}(\mathcal{F}, \mathcal{V})$

rewrite rule $(l \to r)$ is equation $l \approx r$ such that

- $l \notin \mathcal{V}$
- $\mathcal{V}\text{ar}(r) \subseteq \mathcal{V}\text{ar}(l)$

term rewrite system (TRS) is ES all of whose equations are rewrite rules

DEFINITION

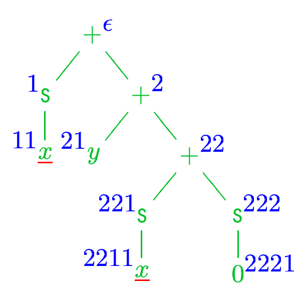
binary relation $\to_{\mathcal{E}}$ on $\mathcal{T}(\mathcal{F}, \mathcal{V})$ for every ES $(\mathcal{F}, \mathcal{E})$:

$$s \to_{\mathcal{E}} t \iff \begin{array}{l} \exists\, p \in \mathcal{P}\text{os}(s) \\ \exists\, l \approx r \in \mathcal{E} \\ \exists \text{ substitution } \sigma \end{array} \quad \text{with} \quad \begin{array}{rcl} s|_p & = & l\sigma \\ t & = & s[r\sigma]_p \end{array} \quad \text{redex}$$

Terms

- signature $\mathcal{F}$ function symbols with arities
- variables $\mathcal{V}$ $\mathcal{F} \cap \mathcal{V} = \varnothing$ infinitely many
- (ground) terms $\mathcal{T}(\mathcal{F}, \mathcal{V})$ $(\mathcal{T}(\mathcal{F}))$

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



not linear

OPERATIONS ON TERMS

- $\mathcal{V}\text{ar}(\,\cdot\,)$ $x \quad y$
- $\mathcal{F}\text{un}(\,\cdot\,)$ $0 \quad \mathsf{s} \quad +$
- $\text{root}(\,\cdot\,)$ $+$
- $|\cdot|$ 10

POSITIONS

- $t|_p$ take subterm of t at position p
- $t[s]_p$ replace subterm in t at position p by s
- $\mathcal{P}\text{os}(t) = \mathcal{P}\text{os}_{\mathcal{F}}(t) \uplus \mathcal{P}\text{os}_{\mathcal{V}}(t)$

DEFINITION

rewrite relation is binary relation R on terms which is closed under contexts and closed under substitutions:

- $s\,R\,t \implies u[s]_p\,R\,u[t]_p$ $\forall$ terms u and positions $p \in \mathcal{P}\text{os}(u)$
- $s\,R\,t \implies s\sigma\,R\,t\sigma$ $\forall$ substitutions σ

LEMMA

relation $\to_{\mathcal{E}}$ is smallest rewrite relation such that $\mathcal{E} \subseteq \to_{\mathcal{E}}$

DERIVED RELATIONS

$\downarrow$ joinability $\downarrow = \to^* \cdot {}^*\!\leftarrow$

$\leftrightarrow^*$ conversion (equivalence relation generated by $\to$)

LEMMA

$\forall$ ES $\mathcal{E}$ $\quad \leftrightarrow^*_{\mathcal{E}} \;=\; \approx_{\mathcal{E}}$ $\quad$ (validity in all models of $\mathcal{E}$)

TERMINOLOGY

- if $s \to^* t$ then s rewrites to t and t is reduct of s
- if $s \to^* u \;{}^*\!\!\leftarrow t$ then u is common reduct of s and t
- if $s \leftrightarrow^* t$ then s and t are convertible
- normal form is term s such that $s \not\to t$ for all t
- $s \to^! t$ if $s \to^* t$ for normal form t

DEFINITION

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

- $\mathcal{R}$ is string rewrite system (SRS) if every $f \in \mathcal{F}$ is unary
- defined symbols $\mathcal{F}_{\mathcal{D}} = \{\, \mathrm{root}(l) \mid l \to r \in \mathcal{R} \,\}$
- constructors $\mathcal{F}_{\mathcal{C}} = \mathcal{F} \setminus \mathcal{F}_{\mathcal{D}}$
- $\mathcal{R}$ is constructor system (CS) if

$$\forall\, f(l_1,\ldots,l_n) \to r \in \mathcal{R} \quad l_1,\ldots,l_n \in \mathcal{T}(\mathcal{F}_{\mathcal{C}},\mathcal{V})$$

constructor terms

PROPERTIES OF TRSs

SN strong normalization termination

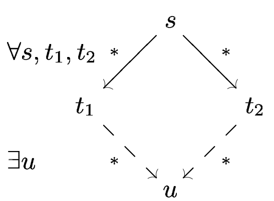
no infinite rewrite sequences

UN unique normal forms

no element has more than one normal form

$\forall s, t_1, t_2$ if $s \to^! t_1$ and $s \to^! t_2$ then $t_1 = t_2$ $\qquad {}^!\!\leftarrow \cdot \to^! \;\subseteq\; =$

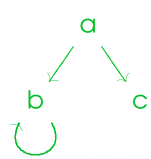
CR confluence Church-Rosser property



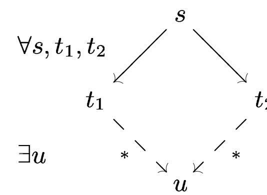
LEMMA

① CR $\Longrightarrow$ UN

② CR $\not\Longleftarrow$ UN



WCR local confluence weak Church-Rosser property



$\leftarrow \cdot \to \;\subseteq\; \downarrow$

LEMMA

① CR $\iff$ $\leftrightarrow^* \subseteq \downarrow$ $\iff$ $\leftrightarrow^* = \downarrow$

② CR $\Longrightarrow$ WCR

③ CR $\not\Longleftarrow$ WCR $\qquad a \leftarrow b \rightleftarrows c \to d$

④ SN $\wedge$ WCR $\Longrightarrow$ CR $\qquad$ Newman's Lemma

WN weak normalization

every element has at least one normal form

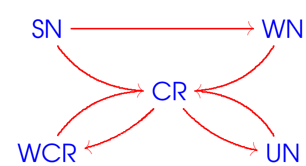
$\forall s\, \exists t \;\; s \to^! t$

LEMMA

① SN $\Longrightarrow$ WN

② SN $\not\Longleftarrow$ WN

③ WN $\wedge$ UN $\Longrightarrow$ CR



semi-completeness CR ∧ WN

every element has unique normal form

completeness CR ∧ SN

diamond property $\diamond$

$$\forall s, t_1, t_2 \quad \exists u \qquad \begin{tikzcd} & s \arrow[dl] \arrow[dr] & \\ t_1 \arrow[dr, dashed] & & t_2 \arrow[dl, dashed] \\ & u & \end{tikzcd} \qquad \leftarrow \cdot \rightarrow \;\subseteq\; \rightarrow \cdot \leftarrow$$

LEMMA

ARS $\mathcal{A} = \langle A, \rightarrow \rangle$ is confluent if $\rightarrow \subseteq \rightarrow_\diamond \subseteq \rightarrow^*$ for some relation $\rightarrow_\diamond$ on A with diamond property

DECIDABILITY

'all' properties of TRSs are undecidable

- SN (even for one-rule TRSs) CR WN ...
- SN is undecidable for confluent TRSs

THEOREM

- CR is decidable for terminating TRSs
- CR is decidable for left-linear right-ground TRSs
- SN is decidable for right-ground TRSs

OPEN PROBLEMS

- is CR decidable for right-ground TRSs ?
- is SN decidable for one-rule SRSs ?

VALIDITY PROBLEM undecidable

instance: ES $(\mathcal{F}, \mathcal{E})$ terms $s, t \in \mathcal{T}(\mathcal{F}, \mathcal{V})$

question: $s \leftrightarrow^*_{\mathcal{E}} t$?

Combinatory Logic

$$\begin{aligned} \mathsf{I} \cdot x &\approx x \\ (\mathsf{K} \cdot x) \cdot y &\approx x \\ ((\mathsf{S} \cdot x) \cdot y) \cdot z &\approx (x \cdot z) \cdot (y \cdot z) \end{aligned}$$

THEOREM

validity problem for ES $(\mathcal{F}, \mathcal{E})$ is decidable if

$\exists$ finite (semi-)complete TRS $(\mathcal{F}, \mathcal{R})$ such that $\underbrace{\leftrightarrow^*_{\mathcal{R}} = \leftrightarrow^*_{\mathcal{E}}}_{\mathcal{R} \text{ implements } \mathcal{E}}$

DECISION PROCEDURE

compute (unique) $\mathcal{R}$-normal forms of s and t and compare:

- yes if equal
- no if different

OVERVIEW

- examples
- term rewriting
- termination
- confluence
- completion
- strategies
- narrowing
- modularity
- further reading

TERMINATION

LEMMA

TRS $\mathcal{R}$ is terminating iff $\exists$ well-founded order $>$ on terms such that

$$s \to_{\mathcal{R}} t \quad \Longrightarrow \quad s > t$$

inconvenient to check all rewrite steps

LEMMA

TRS $\mathcal{R}$ is terminating iff $\exists$ well-founded order $>$ on terms such that

① $l \to r \in \mathcal{R} \quad \Longrightarrow \quad l > r$

② $>$ is closed under contexts

③ $>$ is closed under substitutions

DEFINITION

➜ binary relation $>$ on terms is reduction order if

① closed under contexts

② closed under substitutions

③ proper order (irreflexive and transitive)

④ well-founded

➜ TRS $\mathcal{R}$ and $>$ are compatible if $l > r$ for all $l \to r \in \mathcal{R}$

LEMMA

TRS $\mathcal{R}$ is terminating iff compatible with reduction order

QUESTION

how to construct reduction orders ?

① use algebras (semantic approach)

② use induction (syntactic approach)

DEFINITION

➜ well-founded monotone $\mathcal{F}$-algebra (WFMA) $(\mathcal{A}, >)$ is 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 $a_i > b$

➜ binary relation $>_{\mathcal{A}}$ on terms:

$$s >_{\mathcal{A}} t \quad \Longleftrightarrow \quad \underbrace{[\alpha]_{\mathcal{A}}(s)}_{\text{interpretation of } s \text{ in } \mathcal{A} \text{ under assignment } \alpha} > [\alpha]_{\mathcal{A}}(t) \text{ for all assignments } \alpha$$

➜ TRS $\mathcal{R}$ and WFMA $(\mathcal{A}, >)$ are compatible if $\mathcal{R}$ and $>_{\mathcal{A}}$ are compatible

THEOREM

➜ $>_{\mathcal{A}}$ is reduction order for every WFMA $(\mathcal{A}, >)$

➜ TRS is terminating iff compatible with WFMA

DEFINITION

TRS $\mathcal{R}$ is polynomially terminating if compatible with WFMA $(\mathcal{A}, >)$ such that

① carrier of $\mathcal{A}$ is $\mathbb{N}$

② $>$ is standard order on $\mathbb{N}$

③ $f_{\mathcal{A}}$ is polynomial for every f

$\mathcal{R}$			interpretations		
$x + 0$	$\to$	x	$0_{\mathcal{A}}$	$=$	1
$x + \mathsf{s}(y)$	$\to$	$\mathsf{s}(x + y)$	$\mathsf{s}_{\mathcal{A}}$	$=$	$\lambda x \,.\, x + 1$
$x \times 0$	$\to$	0	$+_{\mathcal{A}}$	$=$	$\lambda xy \,.\, x + 2y$
$x \times \mathsf{s}(y)$	$\to$	$x \times y + x$	$\times_{\mathcal{A}}$	$=$	$\lambda xy \,.\, (x+1)(y+1)^2$

Lexicographic Path Order

DEFINITION

- precedence is proper order $>$ on $\mathcal{F}$
- binary relation $>_{\mathsf{lpo}}$ on terms:
  $s >_{\mathsf{lpo}} t$ if $s = f(s_1,\dots,s_n)$ and either
  ① $t = f(t_1,\dots,t_n)$ and $\exists\, i$
  $$\forall\, j < i \;\; s_j = t_j \qquad s_i >_{\mathsf{lpo}} t_i \qquad \forall\, j > i \;\; s >_{\mathsf{lpo}} t_j$$
  ② $t = g(t_1,\dots,t_m)$ and $f > g$ and $\forall\, j \;\; s >_{\mathsf{lpo}} t_j$
  ③ $\exists\, i \;\; s_i >_{\mathsf{lpo}} t$ or $s_i = t$

THEOREM

$>_{\mathsf{lpo}}$ is reduction order if $>$ is well-founded

Examples

TRS	precedence
$\begin{array}{rcl} 0 + y & \to & y \\ \mathsf{s}(x) + y & \to & \mathsf{s}(x+y) \\ 0 \times y & \to & 0 \\ \mathsf{s}(x) \times y & \to & x \times y + y \end{array}$	$\times > + > \mathsf{s}$
$\begin{array}{rcl} \mathsf{ack}(0,0) & \to & 0 \\ \mathsf{ack}(0,\mathsf{s}(y)) & \to & \mathsf{s}(\mathsf{s}(\mathsf{ack}(0,y))) \\ \mathsf{ack}(\mathsf{s}(x),0) & \to & \mathsf{s}(0) \\ \mathsf{ack}(\mathsf{s}(x),\mathsf{s}(y)) & \to & \mathsf{ack}(x,\mathsf{ack}(\mathsf{s}(x),y)) \end{array}$	$\mathsf{ack} > \mathsf{s}$
$\begin{array}{rclcrcl} \mathsf{e} \cdot x & \to & x & & x \cdot \mathsf{e} & \to & x \\ x^- \cdot x & \to & \mathsf{e} & & x \cdot x^- & \to & \mathsf{e} \\ (x \cdot y) \cdot z & \to & x \cdot (y \cdot z) & & x^{--} & \to & x \\ \mathsf{e}^- & \to & \mathsf{e} & & (x \cdot y)^- & \to & y^- \cdot x^- \\ x^- \cdot (x \cdot y) & \to & y & & x \cdot (x^- \cdot y) & \to & y \end{array}$	${}^- > \cdot > \mathsf{e}$

Lexicographic Path Order: Properties

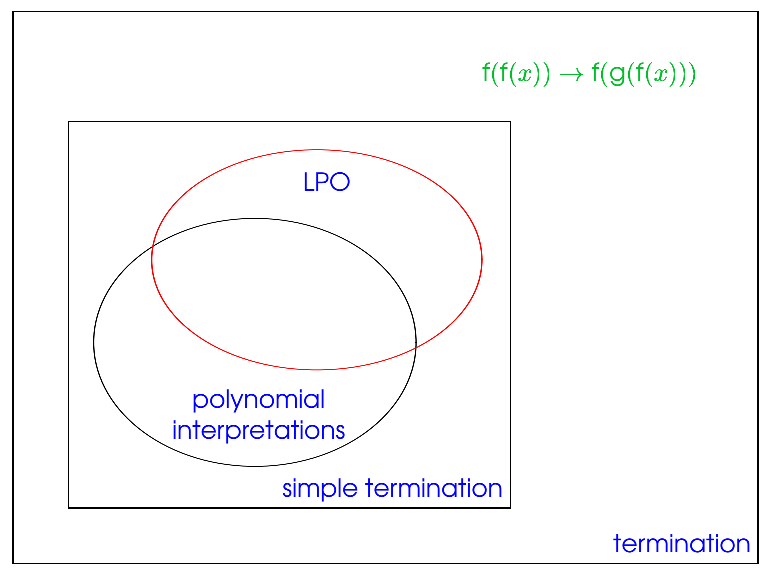
THEOREM

- if $> \subseteq \sqsupset$ then $>_{\mathsf{lpo}} \subseteq \sqsupset_{\mathsf{lpo}}$ (incrementality)
- if $>$ is total then $>_{\mathsf{lpo}}$ is total on ground terms
- following two problems are decidable:

  ① instance: terms s, t precedence $>$
  question: $s >_{\mathsf{lpo}} t$?

  ② instance: terms s, t
  question: $\exists$ precedence $>$ such that $s >_{\mathsf{lpo}} t$?

Comparison



Simple Termination

$\boxed{\text{assumption: signatures are finite}}$

DEFINITION

- binary relation $>$ on terms is simplification order if
  ① closed under contexts
  ② closed under substitutions
  ③ proper order
  ④ subterm property:
  $s[t]_p > t$ for all terms s, t and non-root positions in s
- TRS is simply terminating if compatible with simplification order
- TRS $\mathcal{E}$mb consists of all rewrite rules

$$\mathsf{f}(x_1, \ldots, x_n) \;\to\; x_i$$

- $\trianglelefteq_{\mathsf{emb}} \;=\; \leftarrow^*_{\mathcal{E}\mathsf{mb}}$ (embedding)

Overview

- examples
- term rewriting
- termination
- confluence
- completion
- strategies
- narrowing
- modularity

- further reading

THEOREM

simplification orders are well-founded

proof is based on Kruskal's Tree Theorem:

$\forall$ infinite sequence of ground terms $t_1, t_2, t_3, \ldots$
$\exists\, i < j$ such that $t_i \trianglelefteq_{\mathsf{emb}} t_j$

COROLLARY

simply terminating TRSs are terminating

LEMMA

TRS $\mathcal{R}$ is simply terminating iff $\mathcal{R} \cup \mathcal{E}\mathsf{mb}$ is terminating

EXAMPLE

TRS $\mathsf{f}(\mathsf{f}(x)) \to \mathsf{f}(\mathsf{g}(\mathsf{f}(x)))$ is not simply terminating:

$$\mathsf{f}(\mathsf{f}(x)) \;\to\; \mathsf{f}(\mathsf{g}(\mathsf{f}(x))) \;\to_{\mathcal{E}\mathsf{mb}}\; \mathsf{f}(\mathsf{f}(x))$$

Confluence

Newman's Lemma: SN $\wedge$ WCR $\Rightarrow$ CR

how to prove WCR ?

$$\begin{tikzcd}[column sep=small] & s \arrow[dl] \arrow[dr] & \\ t_1 \arrow[dr, dashed, "*"'] & & t_2 \arrow[dl, dashed, "*"] \\ & u & \end{tikzcd}$$

$\forall s, t_1, t_2$ peak

$\exists u$

$$\begin{aligned} \mathsf{f}(\mathsf{a}, \mathsf{g}(x)) &\to \mathsf{f}(x, x) \\ \mathsf{g}(\mathsf{b}) &\to \mathsf{c} \end{aligned}$$

$$\begin{tikzcd}[column sep=tiny] & \mathsf{f}(\mathsf{g}(\mathsf{b}), \mathsf{g}(\mathsf{b})) \arrow[dl, "1"'] \arrow[dr, "2"] & \\ \mathsf{f}(\mathsf{c}, \mathsf{g}(\mathsf{b})) & & \mathsf{f}(\mathsf{g}(\mathsf{b}), \mathsf{c}) \end{tikzcd}
\quad
\begin{tikzcd}[column sep=tiny] & \mathsf{f}(\mathsf{a}, \mathsf{g}(\mathsf{g}(\mathsf{b}))) \arrow[dl, "21"'] \arrow[dr, "\epsilon"] & \\ \mathsf{f}(\mathsf{a}, \mathsf{g}(\mathsf{c})) & & \mathsf{f}(\mathsf{g}(\mathsf{b}), \mathsf{g}(\mathsf{b})) \end{tikzcd}
\quad
\begin{tikzcd}[column sep=tiny] & \mathsf{f}(\mathsf{a}, \mathsf{g}(\mathsf{b})) \arrow[dl, "2"'] \arrow[dr, "\epsilon"] & \\ \mathsf{f}(\mathsf{a}, \mathsf{c}) & & \mathsf{f}(\mathsf{b}, \mathsf{b}) \end{tikzcd}$$

non-critical non-critical critical

DEFINITION

- overlap is triple $\langle l_1 \to r_1, p, l_2 \to r_2 \rangle$ such that
  1. $l_1 \to r_1$ and $l_2 \to r_2$ are rewrite rules without common variables
  2. $p \in \mathcal{P}os_{\mathcal{F}}(l_1)$
  3. $l_1|_p$ and l_2 are unifiable
  4. if $p = \epsilon$ then $l_1 \to r_1$ and $l_2 \to r_2$ are different

- $$\begin{tikzcd} & l_1\sigma[l_2\sigma]_p = l_1\sigma \arrow[dl, "p"'] \arrow[dr, "\epsilon"] & \\ l_1\sigma[r_2\sigma]_p & \approx & r_1\sigma \end{tikzcd}$$

  σ most general unifier of $l_1|_p$ and l_2

  critical pair

- critical pair $s \approx t$ is convergent if $s \downarrow t$

special case: no critical pairs

WCR (by Critical Pair Lemma) but in general not CR

$$\begin{aligned} f(x,x) &\to a \\ f(x,g(x)) &\to b \\ c &\to g(c) \end{aligned} \qquad \Bigg| \qquad \begin{aligned} f(x,x) &\to a \\ g(x) &\to f(x,g(x)) \\ c &\to g(c) \end{aligned}$$

$$\begin{tikzcd} & f(c,c) \arrow[dl] \arrow[dr, "*"] & \\ a & & b \end{tikzcd} \qquad \begin{tikzcd} c \arrow[r] \arrow[d, "*"'] & g(c) \arrow[r, "*"] & g(a) \\ a & & \end{tikzcd}$$

THEOREM

$\underbrace{\text{left-linear TRSs without critical pairs}}_{\text{orthogonal}}$ are confluent

PROOF

parallel rewriting ($\parallel\!\!\!\rightarrow$) has diamond property

□

CRITICAL PAIR LEMMA

TRS is locally confluent iff all critical pairs are convergent

COROLLARY

terminating TRS is confluent iff all critical pairs are convergent

LEMMA

finite TRSs have finitely many critical pairs

COROLLARY

confluence is decidable for finite terminating TRSs

OVERVIEW

- examples
- term rewriting
- termination
- confluence
- completion
- strategies
- narrowing
- modularity

- further reading

Completion

completion $=$ (compute critical pairs, add new rewrite rules)*

$$\mathcal{R}\qquad \begin{array}{rcll@{\qquad}rcll} x+0 & \to & x & \textcircled{1} & x-s(y) & \to & p(x-y) & \textcircled{4}\\ x-0 & \to & x & \textcircled{2} & p(s(x)) & \to & x & \textcircled{5}\\ x+s(y) & \to & s(x+y) & \textcircled{3} & s(p(x)) & \to & x & \textcircled{6} \end{array}$$

- ➜ SN LPO with precedence $+ > s$, $- > p$
- ➜ WCR ? 4 critical pairs

$$\begin{tikzcd}[column sep=small] & \overline{x+\underline{s(p(y))}} \arrow[dl, "\textcircled{6}"'] \arrow[dr, "\textcircled{3}"] & \\ x+y & & s(x+p(y)) \arrow[ll, bend left, "\textcircled{7}"] \end{tikzcd}
\qquad
\begin{tikzcd}[column sep=small] & \overline{x-\underline{s(p(y))}} \arrow[dl, "\textcircled{6}"'] \arrow[dr, "\textcircled{4}"] & \\ x-y & & p(x-p(y)) \arrow[ll, bend left, "\textcircled{8}"] \end{tikzcd}$$

$$\begin{tikzcd}[column sep=small] & \overline{p(\underline{s(p(x))})} \arrow[dl, "\textcircled{6}"'] \arrow[dr, "\textcircled{5}"] & \\ p(x) & = & p(x) \end{tikzcd}
\qquad
\begin{tikzcd}[column sep=small] & \overline{s(\underline{p(s(x))})} \arrow[dl, "\textcircled{5}"'] \arrow[dr, "\textcircled{6}"] & \\ s(x) & = & s(x) \end{tikzcd}$$

new rewrite rules

$$s(x+p(y)) \to x+y \quad \textcircled{7} \qquad p(x-p(y)) \to x-y \quad \textcircled{8}$$

do not change $\leftrightarrow^*_{\mathcal{R}}$

new critical pairs

$$\begin{tikzcd}[column sep=tiny] & \overline{x+\underline{s(y+p(z))}} \arrow[dl, "\textcircled{7}"'] \arrow[dr, "\textcircled{3}"] & \\ x+(y+z) & & s(x+(y+p(z))) \arrow[dl, "\textcircled{9}"] \\ & s(x+p(y+z)) \arrow[ul, "\textcircled{7}"] & \end{tikzcd}
\qquad
\begin{tikzcd}[column sep=tiny] & \overline{x-\underline{s(y+p(z))}} \arrow[dl, "\textcircled{7}"'] \arrow[dr, "\textcircled{4}"] & \\ x-(y+z) & & p(x-(y+p(z))) \arrow[dl, "\textcircled{9}"] \\ & p(x-p(y+z)) \arrow[ul, "\textcircled{8}"] & \end{tikzcd}$$

$$\begin{tikzcd}[column sep=tiny] & \overline{s(x+\underline{p(y-p(z))})} \arrow[dl, "\textcircled{8}"'] \arrow[dr, "\textcircled{7}"] & \\ s(x+(y-z)) & & x+(y-p(z)) \arrow[dl, "\textcircled{10}"] \\ & x+s(y-z) \arrow[ul, "\textcircled{3}"] & \end{tikzcd}
\qquad
\begin{tikzcd}[column sep=tiny] & \overline{p(x-\underline{p(y-p(z))})} \arrow[dl, "\textcircled{8}"'] \arrow[dr, "\textcircled{8}"] & \\ p(x-(y-z)) & & x-(y-p(z)) \arrow[dl, "\textcircled{10}"] \\ & x-s(y-z) \arrow[ul, "\textcircled{4}"] & \end{tikzcd}$$

new critical pairs

$$\begin{tikzcd}[column sep=small] & \overline{p(\underline{s(x+p(y))})} \arrow[dl, "\textcircled{7}"'] \arrow[dr, "\textcircled{5}"] & \\ p(x+y) & & x+p(y) \arrow[ll, bend left, "\textcircled{9}"] \end{tikzcd}
\qquad
\begin{tikzcd}[column sep=small] & \overline{s(x+\underline{p(s(y))})} \arrow[dl, "\textcircled{5}"'] \arrow[dr, "\textcircled{7}"] & \\ s(x+y) & & x+s(y) \arrow[ll, bend left, "\textcircled{3}"] \end{tikzcd}$$

$$\begin{tikzcd}[column sep=small] & \overline{p(x-\underline{p(s(y))})} \arrow[dl, "\textcircled{5}"'] \arrow[dr, "\textcircled{8}"] & \\ p(x-y) & & x-s(y) \arrow[ll, bend left, "\textcircled{4}"] \end{tikzcd}
\qquad
\begin{tikzcd}[column sep=small] & \overline{s(\underline{p(x-p(y))})} \arrow[dl, "\textcircled{8}"'] \arrow[dr, "\textcircled{6}"] & \\ s(x+y) & & x-p(y) \arrow[ll, bend left, "\textcircled{10}"] \end{tikzcd}$$

new rewrite rules

$$x+p(y) \to p(x+y) \quad \textcircled{9} \qquad x-p(y) \to s(x-y) \quad \textcircled{10}$$

termination is preserved (extend precedence with $+ > p$, $- > s$)

new critical pairs

$$\begin{tikzcd}[column sep=tiny] & \overline{x+\underline{p(s(y))}} \arrow[dl, "\textcircled{5}"'] \arrow[dr, "\textcircled{9}"] & \\ x+y & & p(x+s(y)) \arrow[dl, "\textcircled{3}"] \\ & p(s(x+y)) \arrow[ul, "\textcircled{7}"] & \end{tikzcd}
\qquad
\begin{tikzcd}[column sep=tiny] & \overline{x-\underline{p(s(y))}} \arrow[dl, "\textcircled{5}"'] \arrow[dr, "\textcircled{10}"] & \\ x-y & & s(x-s(y)) \arrow[dl, "\textcircled{4}"] \\ & s(p(x-y)) \arrow[ul, "\textcircled{6}"] & \end{tikzcd}$$

$$\begin{tikzcd}[column sep=tiny] & \overline{x+\underline{p(y-p(z))}} \arrow[dl, "\textcircled{8}"'] \arrow[dr, "\textcircled{9}"] & \\ x+(y-z) & & p(x+(y-p(z))) \arrow[d, "\textcircled{10}"] \\ p(s(x+(y-z))) \arrow[u, "\textcircled{5}"] & & p(x+s(y-z)) \arrow[ll, bend left, "\textcircled{3}"] \end{tikzcd}$$

$$\begin{tikzcd}[column sep=tiny] & \overline{\underline{s(x+p(y))}} \arrow[dl, "\textcircled{9}"'] \arrow[dr, "\textcircled{7}"] & \\ s(p(x+y)) \arrow[rr, bend right, "\textcircled{6}"'] & & x+y \end{tikzcd}
\qquad
\begin{tikzcd}[column sep=tiny] & \overline{\underline{p(x-p(y))}} \arrow[dl, "\textcircled{10}"'] \arrow[dr, "\textcircled{8}"] & \\ p(s(x-y)) \arrow[rr, bend right, "\textcircled{5}"'] & & x-y \end{tikzcd}
\qquad
\begin{tikzcd}[column sep=tiny] & \overline{x-\underline{p(y-p(z))}} \arrow[dl, "\textcircled{10}"'] \arrow[dr, "\textcircled{8}"] & \\ x-p(s(y-z)) \arrow[rr, bend right, "\textcircled{5}"'] & & x-(y-z) \end{tikzcd}$$

$\mathcal{R}$ ① ② ③ ④ ⑤ ⑥

$$
\begin{array}{rcll@{\qquad}rcll}
x+0 & \to & x & ① & x-\mathsf{s}(y) & \to & \mathsf{p}(x-y) & ④ \\
x-0 & \to & x & ② & \mathsf{p}(\mathsf{s}(x)) & \to & x & ⑤ \\
x+\mathsf{s}(y) & \to & \mathsf{s}(x+y) & ③ & \mathsf{s}(\mathsf{p}(x)) & \to & x & ⑥ \\
\mathsf{s}(x+\mathsf{p}(y)) & \to & x+y & ⑦ & x+\mathsf{p}(y) & \to & \mathsf{p}(x+y) & ⑨ \\
\mathsf{p}(x-\mathsf{p}(y)) & \to & x-y & ⑧ & x-\mathsf{p}(y) & \to & \mathsf{s}(x-y) & ⑩
\end{array}
$$

$\mathcal{R}'$ ① ② ③ ④ ⑤ ⑥ ⑦ ⑧ ⑨ ⑩

- all critical pairs of $\mathcal{R}'$ convergent $\implies$ WCR($\mathcal{R}'$)
- SN($\mathcal{R}'$) $\wedge$ WCR($\mathcal{R}'$) $\implies$ $\mathcal{R}'$ is complete
- $\leftrightarrow^*_{\mathcal{R}} \; = \; \leftrightarrow^*_{\mathcal{R}'}$

- simplification can be performed after completion

THEOREM

$\forall$ complete TRS $\mathcal{R}$ $\exists$ complete reduced TRS $\mathcal{R}'$ such that $\leftrightarrow^*_{\mathcal{R}} \; = \; \leftrightarrow^*_{\mathcal{R}'}$

- better idea: perform simplification during completion

THEOREM

if TRSs $\mathcal{R}_1$ and $\mathcal{R}_2$ satisfy

① $\leftrightarrow^*_{\mathcal{R}_1} \; = \; \leftrightarrow^*_{\mathcal{R}_2}$
② $\mathcal{R}_1$ and $\mathcal{R}_2$ are reduced and complete
③ $\mathcal{R}_1$ and $\mathcal{R}_2$ compatible with same reduction order

then

$\mathcal{R}_1 = \mathcal{R}_2$ (modulo variable renaming)

REDUNDANCY

rewrite rules ⑨ and ⑩ make ⑦ and ⑧ redundant

$$
\begin{tikzcd}
\mathsf{s}(x+\mathsf{p}(y)) \arrow[rr, "⑦"] \arrow[dr, "⑨"'] & & x+y \\
& \mathsf{s}(\mathsf{p}(x+y)) \arrow[ur, "⑥"'] &
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathsf{p}(x-\mathsf{p}(y)) \arrow[rr, "⑧"] \arrow[dr, "⑩"'] & & x-y \\
& \mathsf{p}(\mathsf{s}(x-y)) \arrow[ur, "⑤"'] &
\end{tikzcd}
$$

- less rewrite rules $\implies$ less critical pairs
- TRS without redundancy $=$ reduced TRS

DEFINITION

TRS $\mathcal{R}$ is reduced if for all $l \to r \in \mathcal{R}$

① r is normal form with respect to $\mathcal{R}$
② l is normal form with respect to $\mathcal{R} \setminus \{\, l \to r \,\}$

OVERVIEW

- examples
- term rewriting
- termination
- confluence
- completion
- strategies
- narrowing
- modularity

- further reading

Strategies

Definition

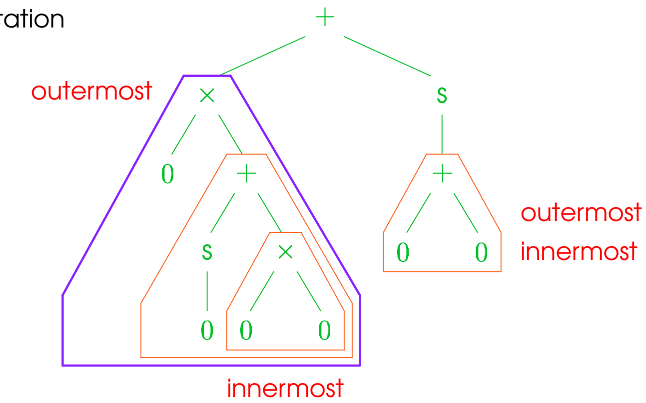
- strategy selects redexes
  - leftmost outermost
  - parallel outermost
- strategy is normalizing if it computes normal forms for all terms that admit normal forms

Lemma

for terminating TRSs every strategy is normalizing

term $\boxed{0 \times \boxed{\mathsf{s}(0) + \boxed{0 \times 0}}} + \mathsf{s}(\boxed{0 + 0})$

tree representation

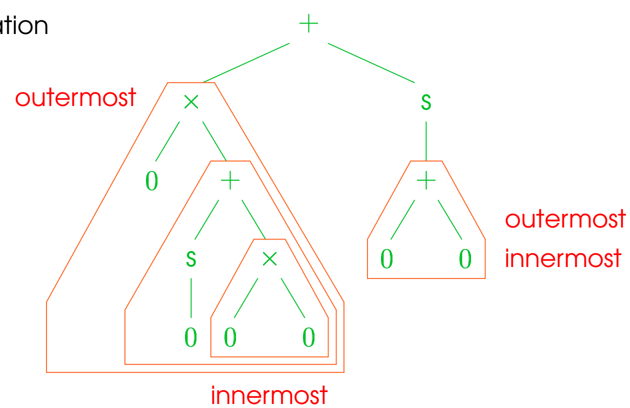


$$\begin{aligned} 0 + y &\to y \\ \mathsf{s}(x) + y &\to \mathsf{s}(x + y) \\ 0 \times y &\to 0 \\ \mathsf{s}(x) \times y &\to x \times y + y \end{aligned}$$

leftmost outermost

term $\boxed{0 \times \boxed{\mathsf{s}(0) + \boxed{0 \times 0}}} + \mathsf{s}(\boxed{0 + 0})$

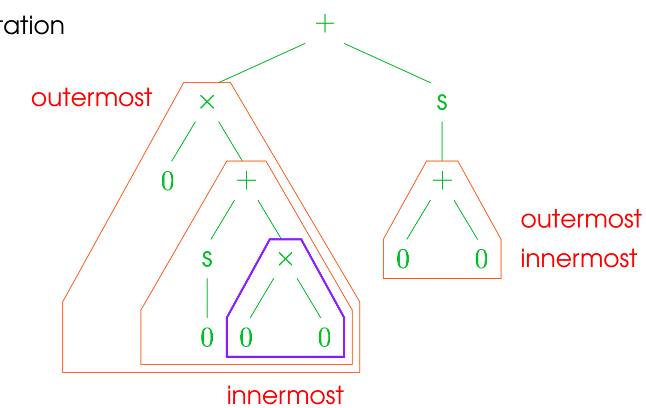
tree representation



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

term $\boxed{0 \times \boxed{\mathsf{s}(0) + \boxed{0 \times 0}}} + \mathsf{s}(\boxed{0 + 0})$

tree representation

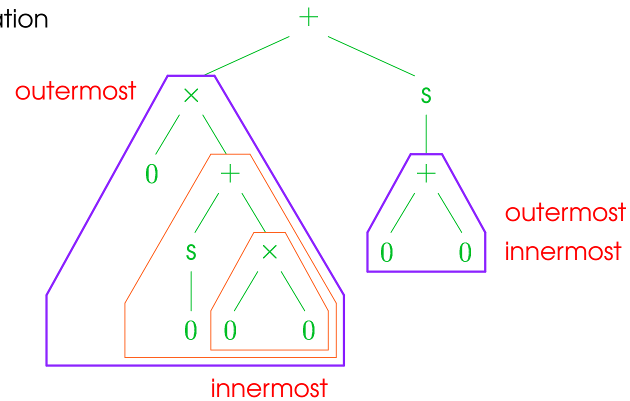


$$\begin{aligned} 0 + y &\to y \\ \mathsf{s}(x) + y &\to \mathsf{s}(x + y) \\ 0 \times y &\to 0 \\ \mathsf{s}(x) \times y &\to x \times y + y \end{aligned}$$

leftmost innermost

term $\boxed{0 \times \boxed{s(0) + \boxed{0 \times 0}}} + s(\boxed{0 + 0})$

tree representation



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

parallel outermost

$\boxed{0 \times (s(0) + (0 \times 0))} + s(0+0) \to \boxed{0 + s(0+0)} \to s(\boxed{0+0}) \to s(0)$

leftmost outermost 3 redexes

$(0 \times (s(0) + \boxed{0 \times 0})) + s(0+0) \to (0 \times \boxed{s(0) + 0}) + s(0+0)$

$\to (0 \times s(\boxed{0+0})) + s(0+0) \to \boxed{0 \times s(0)} + s(0+0) \to 0 + s(\boxed{0+0})$

$\to \boxed{0 + s(0)} \to s(0)$

leftmost innermost 6 redexes

$\boxed{0 \times (s(0) + (0 \times 0))} + s(\boxed{0+0}) \to \boxed{0 + s(0)} \to s(0)$

parallel outermost 3 redexes

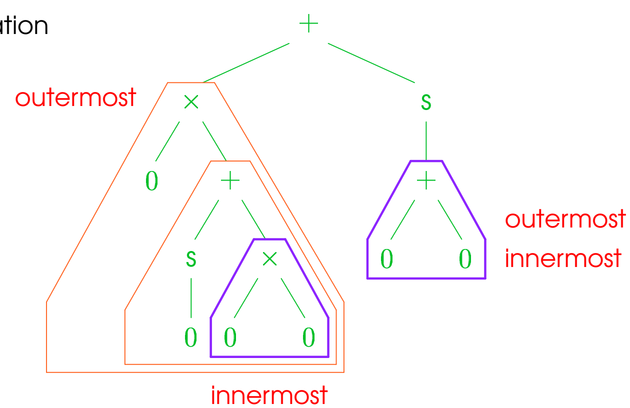
$(0 \times (s(0) + \boxed{0 \times 0})) + s(\boxed{0+0}) \to (0 \times \boxed{s(0) + 0}) + s(0)$

$\to (0 \times s(\boxed{0+0})) + s(0) \to \boxed{0 \times s(0)} + s(0) \to \boxed{0 + s(0)} \to s(0)$

parallel innermost 6 redexes

term $\boxed{0 \times \boxed{s(0) + \boxed{0 \times 0}}} + s(\boxed{0 + 0})$

tree representation



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

parallel innermost

THEOREM

for orthogonal TRSs

- ➜ parallel outermost strategy is normalizing
- ➜ innermost strategies are bad
- ➜ leftmost outermost strategy is not normalizing

$$
\begin{aligned}
a &\to b \\
c &\to c \\
f(x, b) &\to b
\end{aligned}
$$

$f(\underline{c}, a) \to f(\underline{c}, a) \to \cdots$ leftmost outermost

$f(\underline{c}, \underline{a}) \to^* \underline{f(c, b)} \to b$ parallel outermost

THEOREM

leftmost outermost strategy is normalizing for left-normal orthogonal TRSs

no function symbols "after" variables in left-hand sides of rewrite rules

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

easy but important result: Combinatory Logic is left-normal

$$\begin{aligned} \mathsf{I}\,x &\to x \\ \mathsf{K}\,x\,y &\to x \\ \mathsf{S}\,x\,y\,z &\to x\,z\,(y\,z) \end{aligned}$$

THEOREM

for orthogonal TRSs
- every reducible term has needed redex
- needed reduction is normalizing

UNFORTUNATELY

for orthogonal TRSs it is undecidable whether redex is needed

decidable approximations based on powerful tree automata techniques exist

LEMMA

for left-normal orthogonal TRSs leftmost outermost redex is needed

OPTIMALITY

OBSERVATION

parallel outermost is not optimal because it performs useless steps

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

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

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

DEFINITION

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

OVERVIEW

- examples
- term rewriting
- termination
- confluence
- completion
- strategies
- narrowing
- modularity

- further reading

Narrowing

DEFINITION

binary relation $\rightarrowtail_{\mathcal{R}}$ on $\mathcal{T}(\mathcal{F},\mathcal{V})$ for every TRS $(\mathcal{F},\mathcal{R})$:

$$s \overset{\sigma}{\rightarrowtail}_{\mathcal{R}} t \iff \begin{array}{l} \exists\, p \in \mathcal{P}os_{\mathcal{F}}(s) \\ \exists\, l \to r \in \mathcal{R} \\ \exists\, \cancel{\text{substitution}}\ \sigma \\ \quad\ \text{mgu} \end{array} \quad \text{with} \quad \begin{array}{rcl} s|_p\sigma & = & l\sigma \\ t & = & s\sigma[r\sigma]_p \end{array}$$

LEMMA

narrowing is sound for arbitrary TRSs:

$$s \approx t \overset{\sigma}{\rightarrowtail}{}^{*}_{\mathcal{R}_+} \text{true} \implies \sigma \text{ is solution of } s \approx t \quad (s\sigma \leftrightarrow^*_{\mathcal{R}} t\sigma)$$

Example

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

$x + x \approx \mathsf{s}(x)$

$x \mapsto 0$ ↙ ↘ $x \mapsto \mathsf{s}(y)$

$0 \approx \mathsf{s}(0)$ $\qquad$ $\mathsf{s}(y + \mathsf{s}(y)) \approx \mathsf{s}(\mathsf{s}(y))$

failure $\qquad$ $y \mapsto 0$ ↙ ↘ $y \mapsto \mathsf{s}(z)$

$\mathsf{s}(\mathsf{s}(0)) \approx \mathsf{s}(\mathsf{s}(0))$ $\qquad$ $\mathsf{s}(\mathsf{s}(z + \mathsf{s}(\mathsf{s}(z)))) \approx \mathsf{s}(\mathsf{s}(\mathsf{s}(z)))$

↓ $\qquad$ ↙ ↘

true $\qquad$ . . . $\qquad$ . . .

solution $\qquad$ no solutions

$x \mapsto \mathsf{s}(0)$

challenge: reduce search space without losing completeness

Completeness

THEOREM

narrowing is complete for complete TRSs:

$\forall$ solution σ of $s \approx t$ $\exists$ narrowing sequence $s \approx t \overset{\tau}{\rightarrowtail}{}^{*}_{\mathcal{R}_+}$ true

such that $\tau \underbrace{\leqslant_{\mathcal{R}}}_{\text{subsumption (modulo } \mathcal{R})} \sigma\ [\mathcal{V}ar(s \approx t)]$

➜ confluence and termination are essential

	TRS	equation	solution
CR	$\mathsf{a} \to \mathsf{b}$ $\mathsf{a} \to \mathsf{c}$	$\mathsf{b} \approx \mathsf{c}$	ε (empty substitution)
SN	$\mathsf{a} \to \mathsf{f(a)}$	$x \approx \mathsf{f}(x)$	$x \mapsto \mathsf{a}$

➜ termination can be dropped if only $\underbrace{\text{normalized}}_{\sigma(x) \text{ is normal form for every variable } x}$ solutions σ are considered

Completeness

THEOREM

narrowing is complete for complete TRSs:

$\forall$ solution σ of $s \approx t$ $\exists$ narrowing sequence $s \approx t \overset{\tau}{\rightarrowtail}{}^{*}_{\mathcal{R}_+}$ true

such that $\tau \underbrace{\leqslant_{\mathcal{R}}}_{\text{subsumption (modulo } \mathcal{R})} \sigma\ [\mathcal{V}ar(s \approx t)]$

PROOF (LIFTING LEMMA)

① $s\sigma \approx t\sigma \to^* \text{true}$ with σ normalized

$\implies$

② $s \approx t \overset{\tau}{\rightarrowtail}{}^{*} \text{true}$ with $\tau \leqslant \sigma\ [\mathcal{V}ar(s \approx t)]$

① and ② employ same rewrite rules at same positions

☐

Example

$$
\begin{aligned}
0 + y &\to y & 0^2 &\to 0\\
\mathsf{s}(x) + y &\to \mathsf{s}(x+y) & \mathsf{s}(x)^2 &\to x^2 + (\mathsf{s}(x) + x)
\end{aligned}
$$

$$x^2 + x \approx \mathsf{s}(x)$$

$$\rightarrowtail \; x \mapsto \mathsf{s}(y)$$

$$(y^2 + (\mathsf{s}(y) + y)) + \mathsf{s}(y) \approx \mathsf{s}(\mathsf{s}(y))$$

$$y \mapsto 0 \;\swarrow \qquad \searrow$$

$$(0 + (1 + 0)) + 1 \approx 2 \qquad (y^2 + \mathsf{s}(y + y)) + \mathsf{s}(y) \approx \mathsf{s}(\mathsf{s}(y))$$

$$\swarrow \quad \searrow \qquad y \mapsto 0 \;\swarrow \qquad \searrow\; y \mapsto 0$$

$$(1 + 0) + 1 \approx 2 \qquad (0 + \mathsf{s}(0 + 0)) + 1 \approx 2 \qquad (0^2 + 1) + 1 \approx 2$$

$$\searrow \quad \swarrow \qquad \searrow \quad \swarrow$$

$$\mathsf{s}(0 + 0) + 1 \approx 2 \qquad (0 + 1) + 1 \approx 2$$

$$\swarrow \quad \searrow \quad \swarrow$$

$$\mathsf{s}((0 + 0) + 1) \approx 2 \qquad 1 + 1 \approx 2$$

$$\searrow \quad \swarrow$$

$$\mathsf{s}(0 + 1) \approx 2 \;\to\; 2 \approx 2 \;\to\; \text{true}$$

9 different narrowing sequences compute (unique) solution $x \mapsto \mathsf{s}(0)$

Basic Narrowing

DEFINITION

in basic narrowing narrowing steps are not allowed at subterms introduced by previous narrowing substitutions

$$
e\,;\theta \rightarrowtail e';\theta' \quad\Longleftrightarrow\quad
\begin{array}{l}
\exists\, p \in \mathcal{P}\mathrm{os}_{\mathcal{F}}(e)\\
\exists\, l \to r \in \mathcal{R}\\
\exists \text{ mgu } \sigma \text{ of } e\theta|_p \text{ and } l
\end{array}
\quad \text{with} \quad
\begin{aligned}
e' &= e[r]_p\\
\theta' &= \theta\sigma
\end{aligned}
$$

THEOREM

basic narrowing is complete for complete TRSs

NARROWING IS INEFFICIENT

according to Lifting Lemma each rewrite sequence

$$s\sigma \approx t\sigma \;\to^*\; \text{true}$$

corresponds to unique narrowing sequence

$$s \approx t \;\rightarrowtail^*\; \text{true}$$

that computes (generalization of) σ

SOLUTION

- ➜ strategy
  compute only narrowing sequences that corresponds to specific (e.g. leftmost innermost) rewrite sequence
- ➜ rewriting
  rewrite steps can be executed without backtracking

Example

$$
\begin{aligned}
0 + y &\to y & 0^2 &\to 0\\
\mathsf{s}(x) + y &\to \mathsf{s}(x+y) & \mathsf{s}(x)^2 &\to x^2 + (\mathsf{s}(x) + x)
\end{aligned}
$$

$$x^2 + x \approx \mathsf{s}(x)$$

$$\rightarrowtail \; x \mapsto \mathsf{s}(y)$$

$$(y^2 + (\mathsf{s}(y) + y)) + \mathsf{s}(y) \approx \mathsf{s}(\mathsf{s}(y))$$

$$y \mapsto 0 \;\swarrow \qquad \searrow$$

$$(0 + (1 + 0)) + 1 \approx 2 \qquad (y^2 + \mathsf{s}(y + y)) + \mathsf{s}(y) \approx \mathsf{s}(\mathsf{s}(y))$$

$$\swarrow \quad \searrow \qquad y \mapsto 0 \;\swarrow \qquad \searrow\; y \mapsto 0$$

$$(1 + 0) + 1 \approx 2 \qquad (0 + \mathsf{s}(0 + 0)) + 1 \approx 2 \qquad (0^2 + 1) + 1 \approx 2$$

nonbasic $\quad \swarrow \qquad \searrow \quad \swarrow$

$$\mathsf{s}(0 + 0) + 1 \approx 2 \qquad (0 + 1) + 1 \approx 2$$

$\swarrow \quad$ nonbasic $\quad \swarrow$

$$\mathsf{s}((0 + 0) + 1) \approx 2 \qquad 1 + 1 \approx 2$$

nonbasic $\quad \swarrow$

$$\mathsf{s}(0 + 1) \approx 2 \;\to\; 2 \approx 2 \;\to\; \text{true}$$

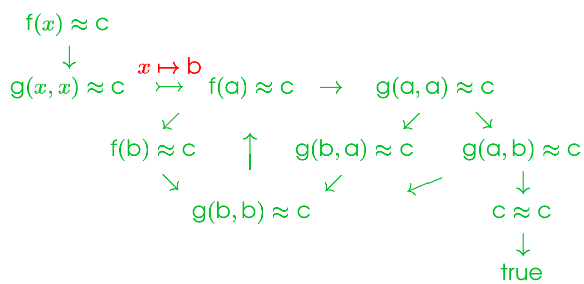
3 different basic narrowing sequences compute solution $x \mapsto \mathsf{s}(0)$

REMARK

termination is essential for the completeness of basic narrowing

$$\begin{array}{lllll} \mathsf{f}(x) \to \mathsf{g}(x,x) & & \mathsf{g}(\mathsf{a},\mathsf{b}) \to \mathsf{c} \\ \mathsf{a} \to \mathsf{b} & & \mathsf{g}(\mathsf{b},\mathsf{b}) \to \mathsf{f}(\mathsf{a}) \end{array}$$

semi-complete



DETERMINISTIC REWRITING

THEOREM

for arbitrary (confluent) TRSs

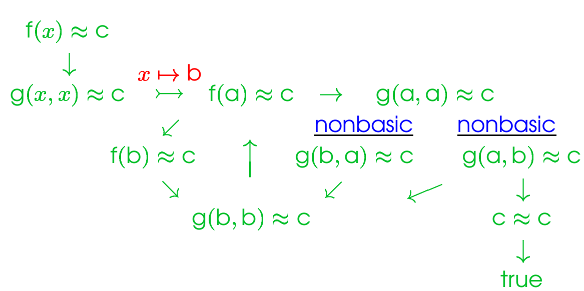
if rewrite step is applicable all other narrowing steps can be ignored without losing completeness

REMARK

termination is essential for the completeness of basic narrowing

$$\begin{array}{lllll} \mathsf{f}(x) \to \mathsf{g}(x,x) & & \mathsf{g}(\mathsf{a},\mathsf{b}) \to \mathsf{c} \\ \mathsf{a} \to \mathsf{b} & & \mathsf{g}(\mathsf{b},\mathsf{b}) \to \mathsf{f}(\mathsf{a}) \end{array}$$

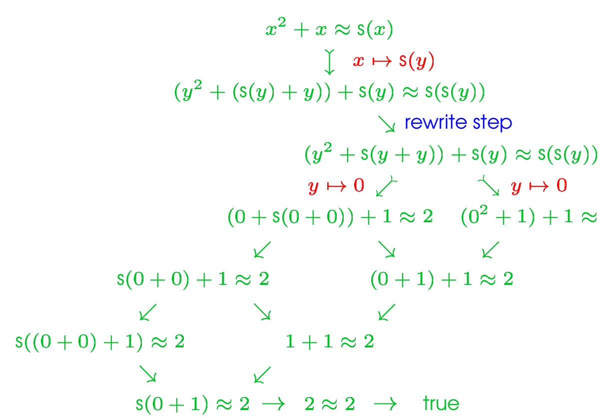
semi-complete



basic narrowing cannot compute solution $x \mapsto \mathsf{b}$

EXAMPLE

$$\begin{array}{lll} 0 + y \to y & & 0^2 \to 0 \\ \mathsf{s}(x) + y \to \mathsf{s}(x+y) & & \mathsf{s}(x)^2 \to x^2 + (\mathsf{s}(x) + x) \end{array}$$



narrowing with deterministic rewriting

Example

$$\begin{aligned} 0 + y &\to y & 0^2 &\to 0 \\ \mathsf{s}(x) + y &\to \mathsf{s}(x + y) & \mathsf{s}(x)^2 &\to x^2 + (\mathsf{s}(x) + x) \end{aligned}$$

$x^2 + x \approx \mathsf{s}(x)$

$\rightsquigarrow$ $x \mapsto \mathsf{s}(y)$

$(y^2 + (\mathsf{s}(y) + y)) + \mathsf{s}(y) \approx \mathsf{s}(\mathsf{s}(y))$

$\searrow$ rewrite step

$(y^2 + \mathsf{s}(y + y)) + \mathsf{s}(y) \approx \mathsf{s}(\mathsf{s}(y))$

$y \mapsto 0$ $\swarrow$ $\searrow$ $y \mapsto 0$

$(0 + \mathsf{s}(0 + 0)) + 1 \approx 2$ $\quad$ $(0^2 + 1) + 1 \approx 2$

rewrite step $\searrow$ $\swarrow$

$(0 + 1) + 1 \approx 2$

$\swarrow$

$1 + 1 \approx 2$

$\swarrow$

$\mathsf{s}(0 + 1) \approx 2 \to 2 \approx 2 \to$ true

narrowing with deterministic rewriting

Overview

- examples
- term rewriting
- termination
- confluence
- completion
- strategies
- narrowing
- modularity

- further reading

Deterministic Rewriting

THEOREM

for arbitrary (confluent) TRSs

if rewrite step is applicable all other narrowing steps can be ignored without losing completeness

DEFINITION

in normal narrowing equations are rewritten to normal form before narrowing steps are computed

COROLLARY

normal narrowing is complete for complete TRSs

Modularity

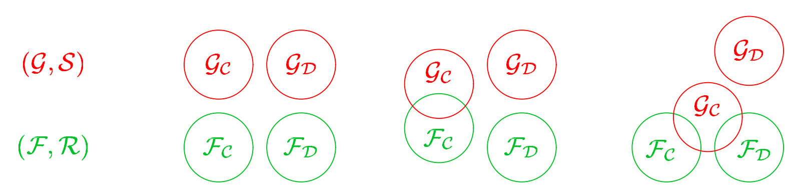
DEFINITION

property of TRSs is modular if it is preserved under union

REMARK

without further restrictions 'no' property of TRSs is modular

termination	a $\to$ b	b $\to$ a
confluence	a $\to$ b	a $\to$ c



two TRSs $\quad$ disjoint $\quad$ constructor-sharing $\quad$ hierarchical

Example

①	$0 + y \to y$ $\mathsf{s}(x) + y \to \mathsf{s}(x + y)$	$0 \times y \to 0$ $\mathsf{s}(x) \times y \to x \times y + y$	②
③	$0 - y \to 0$ $x - 0 \to x$ $\mathsf{s}(x) - \mathsf{s}(y) \to x - y$	$\mathsf{fib}(0) \to \mathsf{s}(0)$ $\mathsf{fib}(\mathsf{s}(0)) \to \mathsf{s}(0)$ $\mathsf{fib}(\mathsf{s}(\mathsf{s}(x))) \to \mathsf{fib}(\mathsf{s}(x)) + \mathsf{fib}(x)$	④
⑤	$\mathsf{nil} +\!\!+ x \to x$ $(x : y) +\!\!+ z \to x : (y +\!\!+ z)$	$0 \div \mathsf{s}(y) \to 0$ $\mathsf{s}(x) \div \mathsf{s}(y) \to \mathsf{s}((x - y) \div \mathsf{s}(y))$	⑥
⑦	$\mathsf{true} \wedge \mathsf{false} \to \mathsf{false}$ $\mathsf{false} \wedge \mathsf{true} \to \mathsf{false}$ $x \wedge x \to x$	$x < 0 \to \mathsf{false}$ $0 < \mathsf{s}(y) \to \mathsf{true}$ $\mathsf{s}(x) < \mathsf{s}(y) \to x < y$	⑧
⑨	$\mathsf{sum}(\mathsf{nil}) \to 0$ $\mathsf{sum}(x : y) \to x + \mathsf{sum}(y)$	$\mathsf{length}(\mathsf{nil}) \to 0$ $\mathsf{length}(x : y) \to \mathsf{s}(\mathsf{length}(y))$	⑩

① ✢ ② ✢ ③ ✢ ④ ✢ ⑤ ✢ ⑥ ✢ ⑦ ✢ ⑧ ✢ ⑨ ✢ ⑩

h cs h d h d cs h cs

REMARK

termination is not modular for disjoint confluent TRSs

$$
\begin{aligned}
\mathsf{f}(\mathsf{a},\mathsf{b},x) &\to \mathsf{f}(x,x,x) \\
\mathsf{a} &\to \mathsf{c} \\
\mathsf{b} &\to \mathsf{c} \\
\mathsf{f}(x,y,z) &\to \mathsf{c}
\end{aligned}
\qquad
\begin{aligned}
\mathsf{g}(x,y,y) &\to x \\
\mathsf{g}(y,y,x) &\to x
\end{aligned}
$$

$$
\begin{aligned}
\mathsf{f}(\mathsf{a},\mathsf{b},\mathsf{g}(\mathsf{a},\mathsf{b},\mathsf{b})) &\to \mathsf{f}(\mathsf{g}(\mathsf{a},\mathsf{b},\mathsf{b}),\mathsf{g}(\mathsf{a},\mathsf{b},\mathsf{b}),\mathsf{g}(\mathsf{a},\mathsf{b},\mathsf{b})) \\
&\to \mathsf{f}(\mathsf{a},\mathsf{g}(\mathsf{a},\mathsf{b},\mathsf{b}),\mathsf{g}(\mathsf{a},\mathsf{b},\mathsf{b})) \\
&\to^+ \mathsf{f}(\mathsf{a},\mathsf{g}(\mathsf{c},\mathsf{c},\mathsf{b}),\mathsf{g}(\mathsf{a},\mathsf{b},\mathsf{b})) \\
&\to \mathsf{f}(\mathsf{a},\mathsf{b},\mathsf{g}(\mathsf{a},\mathsf{b},\mathsf{b}))
\end{aligned}
$$

THEOREM

- termination is modular for disjoint left-linear confluent TRSs
- termination is modular for constructor-sharing confluent CSs

THEOREM

- confluence is modular for disjoint TRSs
- termination is not modular for disjoint TRSs

$$
\mathsf{f}(\mathsf{a},\mathsf{b},x) \to \mathsf{f}(x,x,x)
\qquad
\begin{aligned}
\mathsf{g}(x,y) &\to x \\
\mathsf{g}(x,y) &\to y
\end{aligned}
$$

duplicating collapsing

$$
\begin{aligned}
\mathsf{f}(\mathsf{a},\mathsf{b},\mathsf{g}(\mathsf{a},\mathsf{b})) &\to \mathsf{f}(\mathsf{g}(\mathsf{a},\mathsf{b}),\mathsf{g}(\mathsf{a},\mathsf{b}),\mathsf{g}(\mathsf{a},\mathsf{b})) \\
&\to \mathsf{f}(\mathsf{a},\mathsf{g}(\mathsf{a},\mathsf{b}),\mathsf{g}(\mathsf{a},\mathsf{b})) \\
&\to \mathsf{f}(\mathsf{a},\mathsf{b},\mathsf{g}(\mathsf{a},\mathsf{b}))
\end{aligned}
$$

THEOREM

disjoint union of terminating TRSs $\mathcal{R}$ and $\mathcal{S}$ is terminating if

- $\mathcal{R}$ and $\mathcal{S}$ lack collapsing rules
- $\mathcal{R}$ and $\mathcal{S}$ lack duplicating rules
- $\mathcal{R}$ or $\mathcal{S}$ lacks both collapsing and duplicating rules

THEOREM

- simple termination is modular for constructor-sharing TRSs
- weak normalization is modular for constructor-sharing TRSs
- local confluence is modular for constructor-sharing TRSs

REMARK

confluence is not modular for constructor-sharing TRSs

$$
\begin{aligned}
\mathsf{f}(x,x) &\to \mathsf{a} \\
\mathsf{f}(x,\mathsf{g}(x)) &\to \mathsf{b}
\end{aligned}
\qquad
\mathsf{c} \to \mathsf{g}(\mathsf{c})
$$

$$
\mathsf{a} \leftarrow \mathsf{f}(\mathsf{c},\mathsf{c}) \to \mathsf{f}(\mathsf{c},\mathsf{g}(\mathsf{c})) \to \mathsf{b}
$$

THEOREM

semi-completeness is modular for constructor-sharing TRSs

Further Reading

textbook

F. Baader and T. Nipkow, *Term Rewriting and All That*, Cambridge University Press, 1998.

survey articles

N. Dershowitz and J.-P. Jouannaud, *Rewrite Systems*, in: Handbook of Theoretical Computer Science, Vol. B, North-Holland, pp. 243–320, 1990.

J.W. Klop, *Term Rewriting Systems*, in: Handbook of Logic in Computer Science, Vol. 2, Oxford University Press, pp. 1–116, 1992.

RTA proceedings

LNCS 202, 256, 355, 488, 690, 914, 1103, 1232, 1379, 1631, 1833.

rewriting homepage

`http://rewriting.loria.fr/`

confluence

V. van Oostrom, *Confluence by Decreasing Diagrams*, Theoretical Computer Science 126(2), pp. 259–280, 1994.

V. van Oostrom, *Developing Developments*, Theoretical Computer Science 175(1), pp. 159–181, 1997.

narrowing

S. Antoy, R. Echahed, and M. Hanus, *A Needed Narrowing Strategy*, Journal of the ACM, 2000. To appear.

A. Bockmayri, S. Krischer, and A. Werner, *Narrowing Strategies for Arbitrary Canonical Systems*, Fundamenta Informaticae 24(1,2), pp. 125–155, 1995.

M. Hanus, *The Integration of Functions into Logic Programming: From Theory to Practice*, Journal of Logic Programming 19 & 20, pp. 583–628, 1994.

A. Middeldorp and E. Hamoen, *Completeness Results for Basic Narrowing*, Applicable Algebra in Engineering, Communication and Computing 5, pp. 213–253, 1994.

A. Middeldorp and S. Okui, *A Deterministic Lazy Narrowing Calculus*, Journal of Symbolic Computation 25(6), pp. 733–757, 1998.

modularity

B. Gramlich, *Generalized Sufficient Conditions for Modular Termination of Rewriting*, Applicable Algebra in Engineering, Communication and Computing 5, pp. 131–158, 1994.

M.R.K. Krishna Rao, *Modular Proofs for Completeness of Hierarchical Term Rewriting Systems*, Theoretical Computer Science 151, pp. 487–512, 1995.

E. Ohlebusch, *Modular Properties of Composable Term Rewriting Systems*, Journal of Symbolic Computation 20, pp. 1–41, 1995.

termination

T. Arts and J. Giesl, *Termination of Term Rewriting Using Dependency Pairs*, Theoretical Computer Science 236(1,2), pp. 133–178, 2000.

A. Middeldorp and H. Zantema, *Simple Termination of Rewrite Systems*, Theoretical Computer Science 175(1), pp. 127–158, 1997.

H. Zantema, *Termination of Term Rewriting by Semantic Labelling*, Fundamenta Informaticae 24, pp. 89–105, 1995.

completion

L. Bachmair, *Canonical Equational Proofs*, Birkhäuser, 1991.

L. Bachmair, N. Dershowitz, and D.A. Plaisted, *Completion without Failure*, in: Resolution of Equations in Algebraic Structures, Vol. 2, Academic Press, pp. 1–30, 1989.

strategies

S. Antoy and A. Middeldorp, *A Sequential Reduction Strategy*, Theoretical Computer Science 165(1), pp. 75–95, 1996.

I. Durand and A. Middeldorp, *Decidable Call by Need Computations in Term Rewriting*, Proc. 14th CADE, LNAI 1249, pp. 4–18, 1997.

A. Middeldorp, *Call by Need Computations to Root-Stable Form*, Proc. 24th POPL, pp. 94–105, 1997.

conditional rewriting

E. Ohlebusch, *Transforming Conditional Rewrite Systems with Extra Variables into Unconditional Systems*, Proc. 6th LPAR, LNAI 1705, pp. 111–130, 1999.

T. Suzuki, A. Middeldorp, and T. Ida, *Level-Confluence of Conditional Rewrite Systems with Extra Variables in Right-Hand Sides*, Proc. 6th RTA, LNCS 914, pp. 179–193, 1995.

T. Yamada, J. Avenhaus, C. Loría-Sáenz, and A. Middeldorp, *Logicality of Conditional Rewrite Systems*, Theoretical Computer Science 236(1,2), pp. 209–232, 2000.

tree automata

H. Comon et al., *Tree Automata Techniques and Applications*, `www.grappa.univ-lille3.fr/tata`, 1999.

infinitary rewriting

J.R. Kennaway, J.W. Klop, M.R. Sleep, and F.J. de Vries, *Transfinite Reductions in Orthogonal Term Rewriting Systems*, Information and Computation 119(1), pp. 18–38, 1995.

higher-order rewriting

T. Nipkow and C. Prehofer, *Higher-Order Rewriting and Equational Reasoning*, in: Automated Deduction – A Basis for Applications, Vol. 1, Kluwer, pp. 399–430, 1998.

F. van Raamsdonk, *Higher-Order Rewriting*, Proc. 10th RTA, LNCS 1631, pp. 220–239, 1999.