Efficiently Deciding Uniqueness of Normal Forms and Unique Normalization for Ground TRSs[1]

Bertram Felgenhauer

University of Innsbruck, Austria

5th International Workshop on Confluence
Obergurgl 2016-09-08

[1] supported by FWF project P27528

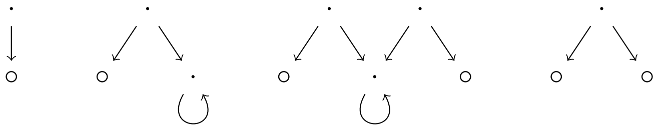
Contents

Definitions and Problem

abstract rewriting

- rewrite relation $\to$, normal forms $\mathsf{NF}(\to)$
- CR: $s \overset{*}{\leftrightarrow} t \implies s \overset{*}{\to} \cdot \overset{*}{\leftarrow} t$
- $\mathsf{UN}^{=}$: $s \overset{*}{\leftrightarrow} t \ \wedge\ s, t \in \mathsf{NF}(\to) \implies s = t$
- $\mathsf{UN}^{\to}$: $s \overset{*}{\leftarrow} \cdot \overset{*}{\to} t \ \wedge\ s, t \in \mathsf{NF}(\to) \implies s = t$

examples

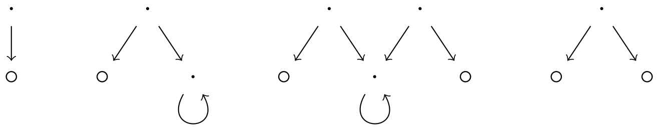


Definitions and Problem

abstract rewriting

- rewrite relation $\to$, normal forms $\mathsf{NF}(\to)$
- CR: $s \overset{*}{\leftrightarrow} t \implies s \overset{*}{\to} \cdot \overset{*}{\leftarrow} t$
- $\mathsf{UN}^{=}$: $s \overset{*}{\leftrightarrow} t \;\wedge\; s, t \in \mathsf{NF}(\to) \implies s = t$
- $\mathsf{UN}^{\to}$: $s \overset{*}{\leftarrow} \cdot \overset{*}{\to} t \;\wedge\; s, t \in \mathsf{NF}(\to) \implies s = t$

examples



questions

- does $\mathsf{UN}^{=}(\mathcal{R})$ hold for a ground TRS $\mathcal{R}$?
- does $\mathsf{UN}^{\to}(\mathcal{R})$ hold for a ground TRS $\mathcal{R}$?

Complexity bounds

$\mathrm{UN}^{=}$

- $\mathcal{O}(||\mathcal{R}||^2 \log ||\mathcal{R}||)$ (Verma, Rusinovich, Lugiez, 2001)
- here: $\mathcal{O}(||\mathcal{R}|| \log ||\mathcal{R}||)$

$\mathrm{UN}^{\rightarrow}$

- PTIME (Verma 2009; Godoy, Jaquemard 2009)
- here: $\mathcal{O}(||\mathcal{R}||^3)$

Contents

Procedure for $\mathrm{UN}^=$

1. currying
2. recognize normal forms
3. congruence closure
4. profit

Currying

example

$\mathcal{R}$: $\mathsf{h}(\mathsf{c},\mathsf{c}) \to \mathsf{f}(\mathsf{c}) \quad \mathsf{a} \to \mathsf{f}(\mathsf{b}) \quad \mathsf{f}(\mathsf{b}) \to \mathsf{b} \quad \mathsf{h}(\mathsf{b},\mathsf{a}) \to \mathsf{c} \quad \mathsf{c} \to \mathsf{c}$

$\mathcal{R}^\circ$: $\mathsf{h} \circ \mathsf{c} \circ \mathsf{c} \to \mathsf{f} \circ \mathsf{c} \quad \mathsf{a} \to \mathsf{f} \circ \mathsf{b} \quad \mathsf{f} \circ \mathsf{b} \to \mathsf{b} \quad \mathsf{h} \circ \mathsf{b} \circ \mathsf{a} \to \mathsf{c} \quad \mathsf{c} \to \mathsf{c}$

Procedure for $\mathsf{UN}^=$

1. currying
   - preserves and reflects $\mathsf{UN}^=$
   - bounds arity and simplifies later algorithms
2. recognize normal forms
3. congruence closure
4. profit

Recognize Normal Forms

deterministic bottom-up tree automaton recognizing normal forms

- only final states
- one state $[s]$ for each normal form subterm $s \trianglelefteq \mathcal{R}^\circ$
- one extra state $[\star]$ ($\star$ fresh constant)
- $f([s_1], \dots, [s_n]) \to [f(s_1, \dots, s_n)]$ for $[f(s_1, \dots, s_n)] \neq [\star]$
- $f([s_1], \dots, [s_n]) \to [\star]$ for $f(s_1, \dots, s_n) \not\trianglelefteq \mathcal{R}^\circ$

Example: Recognize Normal Forms

$\mathcal{R}^{\circ}$: $\mathsf{h} \circ \mathsf{c} \circ \mathsf{c} \to \mathsf{f} \circ \mathsf{c}$ $\quad \mathsf{a} \to \mathsf{f} \circ \mathsf{b}$ $\quad \mathsf{f} \circ \mathsf{b} \to \mathsf{b}$ $\quad \mathsf{h} \circ \mathsf{b} \circ \mathsf{a} \to \mathsf{c}$ $\quad \mathsf{c} \to \mathsf{c}$

automaton

- $Q = Q_f = \{[\star], [\mathsf{b}], [\mathsf{f}], [\mathsf{h}], [\mathsf{hb}], [\mathsf{hc}]\}$

$$\begin{array}{ccc} \mathsf{b} \to [\mathsf{b}] & \mathsf{f} \to [\mathsf{f}] & \mathsf{h} \to [\mathsf{h}] \\ [\mathsf{h}] \circ [\mathsf{b}] \to [\mathsf{hb}] & [\mathsf{f}] \circ [\mathsf{c}] \to [\mathsf{hc}] & \ldots \to [\star] \end{array}$$

- $\ldots$: all but a, b, c, f, $[\mathsf{h}] \circ [\mathsf{b}]$, $[\mathsf{f}] \circ [\mathsf{c}]$, $[\mathsf{f}] \circ [\mathsf{b}]$, $[\mathsf{hc}] \circ [\mathsf{c}]$
(32 transitions, but only 8 exceptions)

Procedure for $\mathsf{UN}^=$

1. currying
   - preserves and reflects $\mathsf{UN}^=$
   - bounds arity and simplifies later algorithms
2. recognize normal forms
   - deterministic tree automaton $\mathcal{N}$ without non-accepting states
   - implicit form of linear size (explicit: quadratic size)
3. congruence closure
4. profit

Congruence Closure

purpose

- decide $s \xleftrightarrow[\mathcal{R}^\circ]{*} t$

symmetric ground tree transducer $\mathcal{C}$ recognizing convertible terms

- states $[s]_{\mathcal{R}^\circ} = \{t \mid s \xleftrightarrow[\mathcal{R}^\circ]{*} t\}$ for $s \trianglelefteq \mathcal{R}^\circ$
- $f([s_1]_{\mathcal{R}^\circ}, \ldots, [s_n]_{\mathcal{R}^\circ}) \to [f(s_1, \ldots, s_n)]_{\mathcal{R}^\circ}$ for $f(s_1, \ldots, s_n) \trianglelefteq \mathcal{R}^\circ$

notes

- can computed is efficiently: $\mathcal{O}(||\mathcal{R}^\circ|| \log ||\mathcal{R}^\circ||)$ time
- $s \xleftrightarrow[\mathcal{R}^\circ]{*} t$ iff $s{\downarrow_{\mathcal{C}}} = t{\downarrow_{\mathcal{C}}}$
- transitions are deterministic

Example: Congruence Closure (1/2)

$\mathcal{R}^\circ$: $\quad \mathsf{h} \circ \mathsf{c} \circ \mathsf{c} \to \mathsf{f} \circ \mathsf{c} \quad \mathsf{a} \to \mathsf{f} \circ \mathsf{b} \quad \mathsf{f} \circ \mathsf{b} \to \mathsf{b} \quad \mathsf{h} \circ \mathsf{b} \circ \mathsf{a} \to \mathsf{c} \quad \mathsf{c} \to \mathsf{c}$

congruence classes

$$\begin{aligned} A &= [\mathsf{a}] = [\mathsf{f} \circ \mathsf{b}] = [\mathsf{b}] & F &= [\mathsf{f}] & H &= [\mathsf{h}] \\ C &= [\mathsf{c}] = [\mathsf{h} \circ \mathsf{b} \circ \mathsf{a}] & H_1 &= [\mathsf{h} \circ \mathsf{c}] \\ F_1 &= [\mathsf{f} \circ \mathsf{c}] = [\mathsf{h} \circ \mathsf{c} \circ \mathsf{c}] & H_2 &= [\mathsf{h} \circ \mathsf{b}] \end{aligned}$$

automaton

$$\begin{array}{llll} a \to A & f \to F & H \circ C \to H_1 & H_1 \circ C \to F_1 \\ b \to A & h \to H & H \circ A \to H_2 & H_2 \circ A \to C \\ c \to C & & F \circ A \to A & F \circ C \to F_1 \end{array}$$

Example: Congruence Closure (2/2)

automaton

$$
\begin{array}{cccc}
a \to A & f \to F & H \circ C \to H_1 & H_1 \circ C \to F_1 \\
b \to A & h \to H & H \circ A \to H_2 & H_2 \circ A \to C \\
c \to C & & F \circ A \to A & F \circ C \to F_1
\end{array}
$$

example

$$
\begin{aligned}
\mathsf{h} \circ \mathsf{b} \circ \mathsf{b} &\xrightarrow{*} H \circ A \circ A \to H_2 \circ A \to C \\
\mathsf{f} \circ (\mathsf{h} \circ (\mathsf{h} \circ \mathsf{b} \circ \mathsf{b}) \circ (\mathsf{h} \circ \mathsf{b} \circ \mathsf{b})) &\xrightarrow{*} F \circ (H \circ C \circ C) \xrightarrow{*} {\color{red} F \circ F_1} \\
\mathsf{f} \circ (\mathsf{f} \circ (\mathsf{h} \circ \mathsf{b} \circ \mathsf{b})) &\xrightarrow{*} F \circ (F \circ C) \to {\color{red} F \circ F_1}
\end{aligned}
$$

$$
\Rightarrow \; \mathsf{f} \circ (\mathsf{h} \circ (\mathsf{h} \circ \mathsf{b} \circ \mathsf{b}) \circ (\mathsf{h} \circ \mathsf{b} \circ \mathsf{b})) \xleftrightarrow[\mathcal{R}^\circ]{*} \mathsf{f} \circ (\mathsf{f} \circ (\mathsf{h} \circ \mathsf{b} \circ \mathsf{b}))
$$

Procedure for $\mathrm{UN}^=$

1. currying
   - preserves and reflects $\mathrm{UN}^=$
   - bounds arity and simplifies later algorithms
2. recognize normal forms
   - deterministic tree automaton $\mathcal{N}$ without non-accepting states
   - implicit form of linear size (explicit: quadratic size)
3. congruence closure
   - deterministic tree automaton $\mathcal{C}$ representing convertible terms
4. profit

Decide UN$^=$

procedure

1. enumerate runs $s \to^* (q, q')$ in the product automaton $\mathcal{C} \times \mathcal{N}$
2. fail when $s \to^* (q, q'_1)$ and $t \to^* (q, q'_2)$ for distinct terms s and t
3. succeed when all runs are exhausted

justification

- in step 2, s and t are convertible normal forms, hence UN$^=$ fails
- note that UN$^=$ holds iff $s = t$ for normal forms s and t such that $s \overset{*}{\leftrightarrow} t$ with a root step
- in step 3, we have shown that all subterms of $\mathcal{R}$ have at most one convertible normal form, hence UN$^=$ holds

Example: Decide UN$^=$

$$
\begin{array}{llllll}
\mathcal{C}: & a \to A & f \to F & H \circ C \to H_1 & H_1 \circ C \to F_1 \\
& b \to A & h \to H & H \circ A \to H_2 & H_2 \circ A \to C \\
& c \to C & & F \circ A \to A & F \circ C \to F_1 \\
\mathcal{N}: & & \mathsf{f} \to [\mathsf{f}] & [\mathsf{h}] \circ [\mathsf{b}] \to [\mathsf{hb}] & [\mathsf{f}] \circ [\mathsf{c}] \to [\mathsf{hc}] \\
& \mathsf{b} \to [\mathsf{b}] & \mathsf{h} \to [\mathsf{h}] & \ldots \to [\star] &
\end{array}
$$

$\ldots$: all but a, b, c, f, $[\mathsf{h}] \circ [\mathsf{b}]$, $[\mathsf{f}] \circ [\mathsf{c}]$, $[\mathsf{f}] \circ [\mathsf{b}]$, $[\mathsf{hc}] \circ [\mathsf{c}]$

1: $\mathsf{b} \to (A, [\mathsf{b}])$

2: $\mathsf{f} \to (F, [\mathsf{f}])$

2: $\mathsf{h} \to (H, [\mathsf{h}])$

4: $(H, [\mathsf{h}]) \circ (A, [\mathsf{b}]) \to (H_2, [\mathsf{hb}])$

5: $(H_2, [\mathsf{hb}]) \circ (A, [\mathsf{b}]) \to (C, [\star])$

6: $(F, [\mathsf{f}]) \circ (C, [\star]) \to (F_1, [\star])$

7: $(H, [\mathsf{h}]) \circ (C, [\star]) \to (H_1, [\star])$

8: $(H_1, [\star]) \circ (C, [\star]) \to (F_1, [\star])$

Example: Profit!

$\mathcal{R}$: $\mathsf{h}(\mathsf{c},\mathsf{c}) \to \mathsf{f}(\mathsf{c}) \quad \mathsf{a} \to \mathsf{f}(\mathsf{b}) \quad \mathsf{f}(\mathsf{b}) \to \mathsf{b} \quad \mathsf{h}(\mathsf{b},\mathsf{a}) \to \mathsf{c} \quad \mathsf{c} \to \mathsf{c}$

- this GTRS is not $\mathrm{UN}^{=}$
- witness for $\mathcal{R}^\circ$: $h \circ (h \circ b \circ b) \circ (h \circ b \circ b) \xleftrightarrow[\mathcal{R}^\circ]{*} f \circ (h \circ b \circ b)$
- witness for $\mathcal{R}$: $h(h(b,b),h(b,b)) \xleftrightarrow[\mathcal{R}]{*} f(h(b,b))$

Procedure for $\mathsf{UN}^=$

1. currying
   - preserves and reflects $\mathsf{UN}^=$
   - bounds arity and simplifies later algorithms
2. recognize normal forms
   - deterministic tree automaton $\mathcal{N}$ without non-accepting states
   - implicit form of linear size (explicit: quadratic size)
3. congruence closure
   - deterministic tree automaton $\mathcal{C}$ representing convertible terms
4. profit
   - decide $\mathsf{UN}^=$
   - everything can be computed in $\mathcal{O}(||\mathcal{R}|| \log ||\mathcal{R}||)$ time
   - implemented in CSI

Contents

Summary

done

- quasi-linear algorithm for deciding $\mathrm{UN}^{=}$
  - improving known result by Verma
- key: congruence closure produces deterministic tree automata
  - has been observed before by Bachmair et al.
- $\mathrm{UN}^{\rightarrow}$ for ground TRSs (cubic time by different method)

todo

- NFP $(s \xleftrightarrow{*} t \;\wedge\; s \in \mathrm{NF}(\rightarrow) \implies s \xrightarrow{*} t)$
- formalization
- non-ground systems?

Summary

done

- quasi-linear algorithm for deciding $\mathrm{UN}^{=}$
  - improving known result by Verma
- key: congruence closure produces deterministic tree automata
  - has been observed before by Bachmair et al.
- $\mathrm{UN}^{\rightarrow}$ for ground TRSs (cubic time by different method)

todo

- NFP $(s \xleftrightarrow{*} t \;\wedge\; s \in \mathrm{NF}(\rightarrow) \implies s \xrightarrow{*} t)$
- formalization
- non-ground systems?

Thanks!

Literature

- Verma, R., Rusinowitch, M., Lugiez, D.:
  Algorithms and reductions for rewriting problems.
  FI 46(3), 257–276 (2001)
- Nieuwenhuis, R., Oliveras, A.:
  Fast congruence closure and extensions.
  I&C 205(4), 557–580 (2007)
- Verma, R.:
  Complexity of normal form properties and reductions for term rewriting problems.
  FI 92(1–2), 145–168 (2009)
- Bachmair, L., Tiwari, A., Vigneron, L.:
  Abstract congruence closure.
  JAR 31, 129–168 (2003)