Deciding Confluence of Certain Term Rewriting Systems in Polynomial Time

Ashish Tiwari

{**tiwari**}**@csl.sri.com**

http://www.csl.sri.com/.

Computer Science Laboratory

SRI International

333 Ravenswood

Menlo Park, CA 94025

An Abstract Model

Term rewrite systems

- a binary relation on the set of terms.
- useful abstraction to study variety of things, functional programs, etc.
- are defined by rules, each of which says when a certain term can be replaced by another.

Example. A TRS specified by two rules.

$$\begin{aligned} f(2n+1) &\rightarrow f(3*(2n+1)+1) \\ f(2n) &\rightarrow f(n) \end{aligned}$$

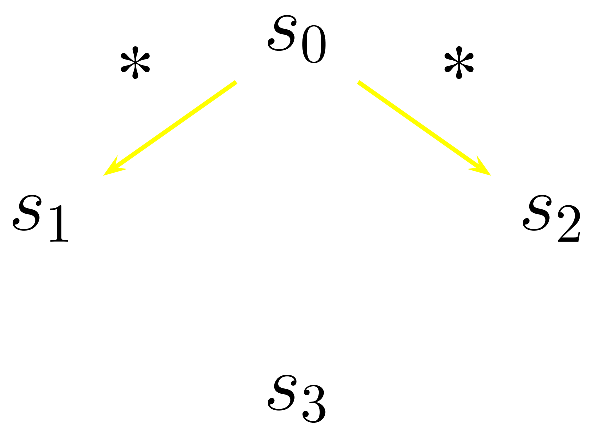
Properties of Binary Relations

Two main properties of interest for a binary relation $\rightarrow$:

- Termination: Starting from some element, do we always hit a dead end?

$$s_0 \longrightarrow s_1 \longrightarrow s_2 \longrightarrow \cdots$$

- Confluence: In case of a choice, if we take different paths, do we always have the option of meeting again?



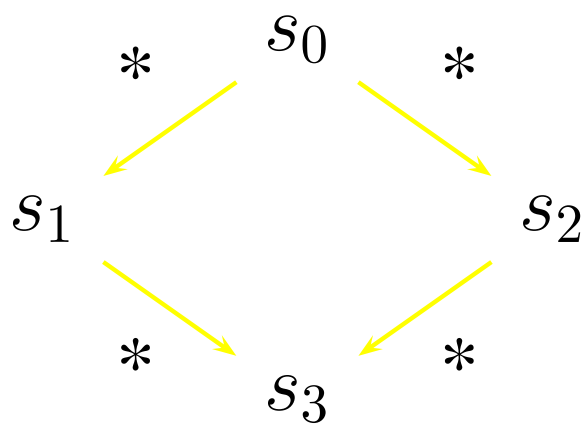
Properties of Binary Relations

Two main properties of interest for a binary relation $\rightarrow$:

- Termination: Starting from some element, do we always hit a dead end?

$$s_0 \longrightarrow s_1 \longrightarrow s_2 \longrightarrow \cdots$$

- Confluence: In case of a choice, if we take different paths, do we always have the option of meeting again?



Motivation

- For terminating systems, confluence implies uniqueness of normal forms. Thus, all choice points can be treated as "don't-care" choices.
- Confluence implies at most one normal form for any term. Thus, a partial function from a term to its normal form is well-defined if the rewrite relation is confluent.
- Rewrite rules, and their ground instances in particular, are often used for simplification in theorem proving tasks. Often, little is provably known about the library of all rules. In such cases it helps to know if the used instances are confluent and terminating.

Examples

Consider the ground rewrite system

$$\mathbb{R}_0 \;=\; \{a \rightarrow fab,\; fab \rightarrow fba\}$$

The terms fba and $f(fba)b$ are congruent modulo $\mathbb{R}_0$.

$$fba \;\leftarrow\; fab \;\rightarrow\; f(fab)b \;\rightarrow\; f(fba)b$$

But are they both reducible to a common term?

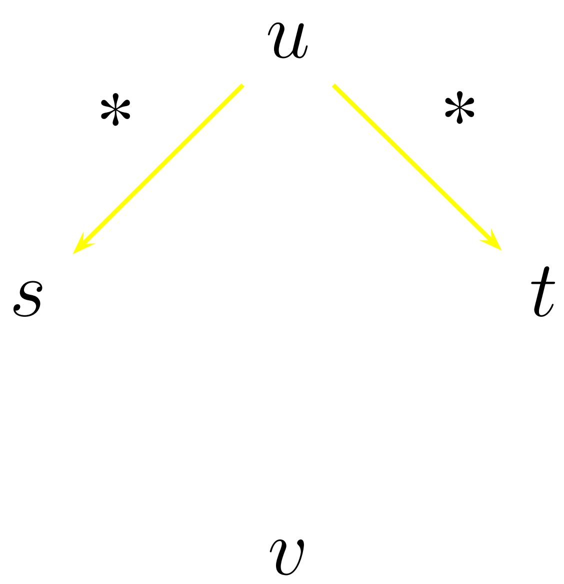
$$\begin{array}{rcccccc} fba & \rightarrow & fb(fab) & \rightarrow & fb(fba) & \rightarrow & \cdots \\ f(fba)b & \rightarrow & f(fb(fab))b & \rightarrow & f(fb(fba))b & \rightarrow & \ldots \end{array}$$

No!

Definition of Confluence

$\mathbb{R}$	:	Finite set of directed ground equations
$s \rightarrow_{\mathbb{R}} t$	:	if $s = s[l]$ and $t = s[r]$ for some $l \rightarrow r \in \mathbb{R}$
$s \rightarrow^*_{\mathbb{R}} t$	:	Transitive closure of $\rightarrow_{\mathbb{R}}$
$s \leftrightarrow^*_{\mathbb{R}} t$	:	Symmetric-transitive closure of $\rightarrow_{\mathbb{R}}$

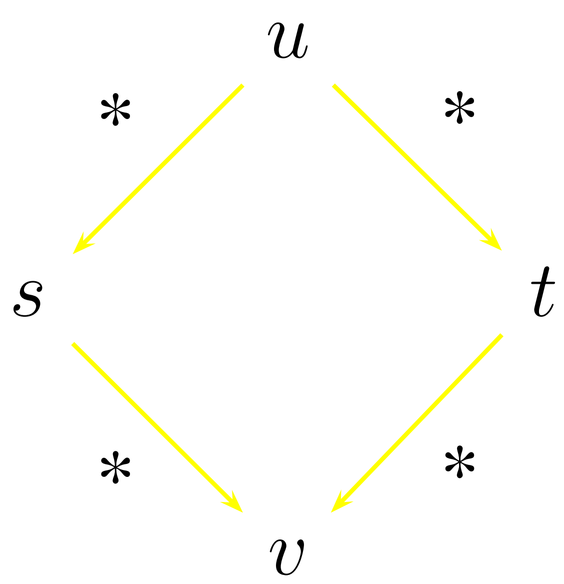
$\mathbb{R}$ is (ground) confluent if $\forall u, s, t \in \mathcal{T}(\Sigma)$



Definition of Confluence

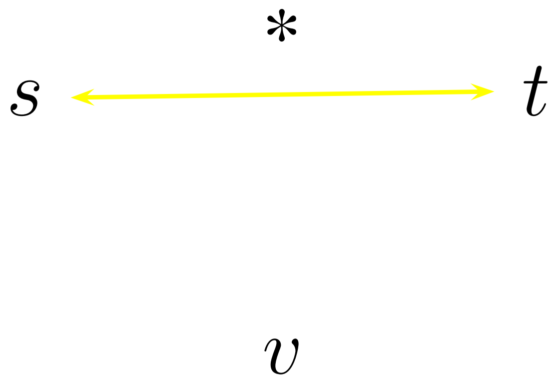
$\mathbb{R}$	:	Finite set of directed ground equations
$s \rightarrow_{\mathbb{R}} t$	:	if $s = s[l]$ and $t = s[r]$ for some $l \rightarrow r \in \mathbb{R}$
$s \rightarrow^*_{\mathbb{R}} t$	:	Transitive closure of $\rightarrow_{\mathbb{R}}$
$s \leftrightarrow^*_{\mathbb{R}} t$	:	Symmetric-transitive closure of $\rightarrow_{\mathbb{R}}$

$\mathbb{R}$ is (ground) confluent if $\forall u, s, t \in \mathcal{T}(\Sigma)$



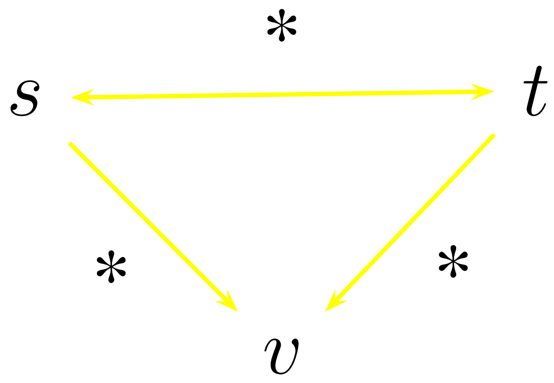
Simple Results

- Equivalent definition of confluence: $\mathbb{R}$ is (ground) confluent if $\forall s, t \in \mathcal{T}(\Sigma)$



Simple Results

- Equivalent definition of confluence: $\mathbb{R}$ is (ground) confluent if $\forall s, t \in \mathcal{T}(\Sigma)$



- For terminating systems, confluence is equivalent to local confluence: check confluence for "local peaks".

Some History

- Confluence decidable for ground systems [DHLT:LICS1987] and [O:TCS1987]: ground tree transducers
- Existence of polynomial time decision procedure open for many years.
- Confluence decidable in poly time for ground systems over one unary symbol [LICS2001 short presentation]
- Confluence decidable in poly time for ground systems [CGN:FOCS2001]

In this paper:

- give a poly time procedure for this problem (again!)
- poly time procedure to decide confluence of certain non-ground rewrite systems

The Crucial Relations

That need to be "computed":

- Congruence relation: $\leftrightarrow^*_{\mathbb{R}}$
  Decided using Congruence closure algorithms
- Reachability relation: $\longrightarrow^*_{\mathbb{R}}$
  Decided using Ground Tree Transducers
- Joinability relation: $\longrightarrow^*_{\mathbb{R}} \circ \longleftarrow^*_{\mathbb{R}}$
  Compose two GTTs

Checking confluence

$$\leftrightarrow^* \subseteq \longrightarrow^* \circ \longleftarrow^*$$

reduces to language inclusion problem for tree automata.
But that is EXPTIME.

Abstraction

Transform ground TRS $\mathbb{R}$ over Σ to a flat TRS over $\Sigma \cup K$.

$$\frac{\mathbb{R} \cup \{s[u] \to t\}}{\mathbb{R} \cup \{s[c] \to t,\ u \to c\}}$$

$$\frac{\mathbb{R} \cup \{s \to t[u]\}}{\mathbb{R} \cup \{s \to t[c],\ c \to u\}}$$

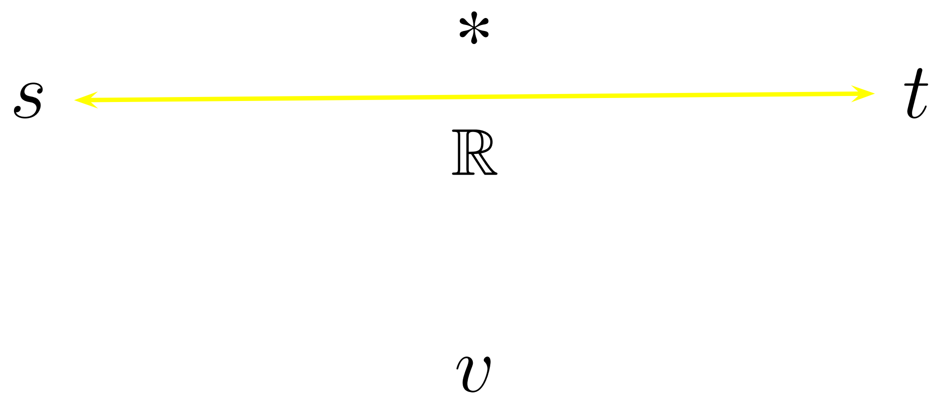
$\therefore$ wlog each rule in $\mathbb{R}$ is either flat or the inverse of a flat rule.

$$f(c_1, \ldots, c_n) \to c, \qquad c \to d$$

Abstract Congruence Closure

First idea was to

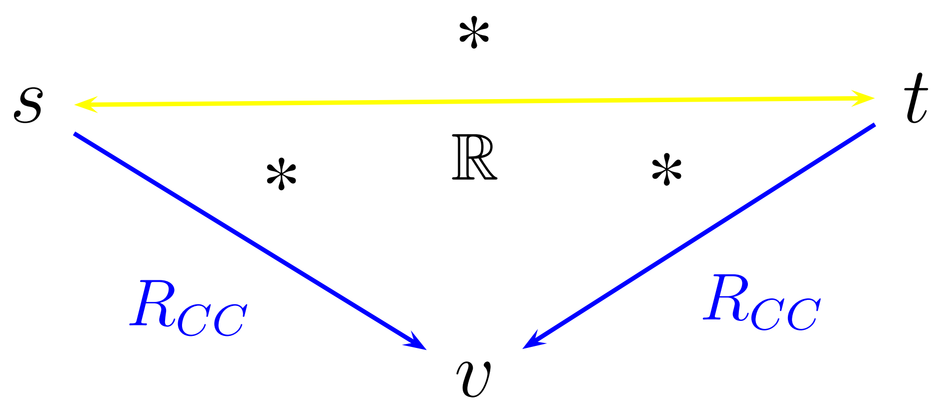
- Describe the congruence relation induced by $\mathbb{R}$ by a convergent flat rewrite system R_{CC}: abstract congruence closure.



Abstract Congruence Closure

First idea was to

- Describe the congruence relation induced by $\mathbb{R}$ by a convergent flat rewrite system R_{CC}: abstract congruence closure.

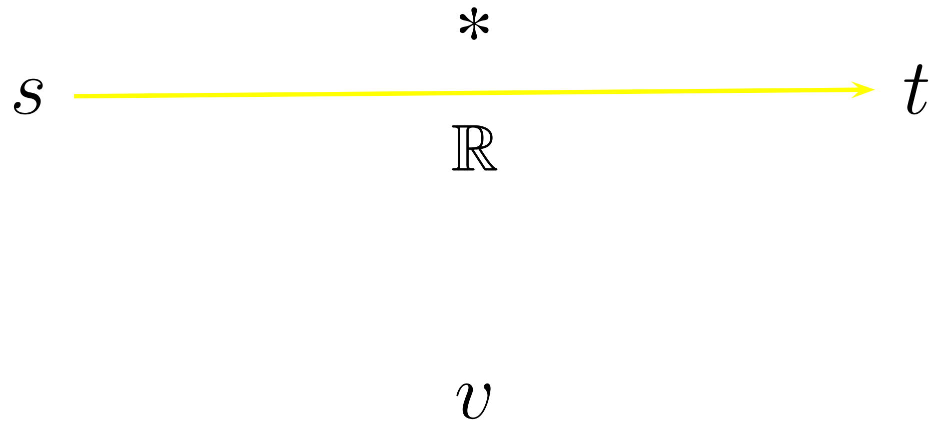


- The relation $\rightarrow_{R_{CC}}$ is terminating.
- Standard completion on a flat $\mathbb{R}$ suffices.

Abstract Rewrite Closure

Second idea was to

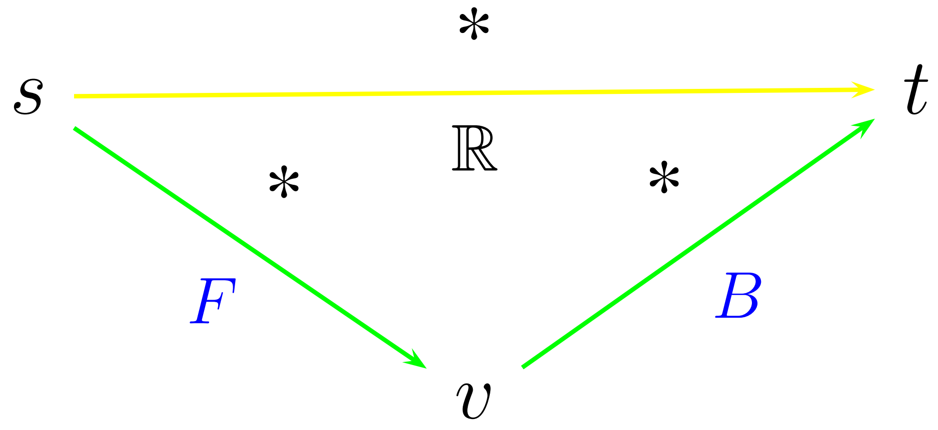
- Describe the reachability relation induced by $\mathbb{R}$ by an "asymmetric convergent flat rewrite system" (F, B): abstract rewrite closure.



Abstract Rewrite Closure

Second idea was to

- Describe the reachability relation induced by $\mathbb{R}$ by an "asymmetric convergent flat rewrite system" (F, B): abstract rewrite closure.



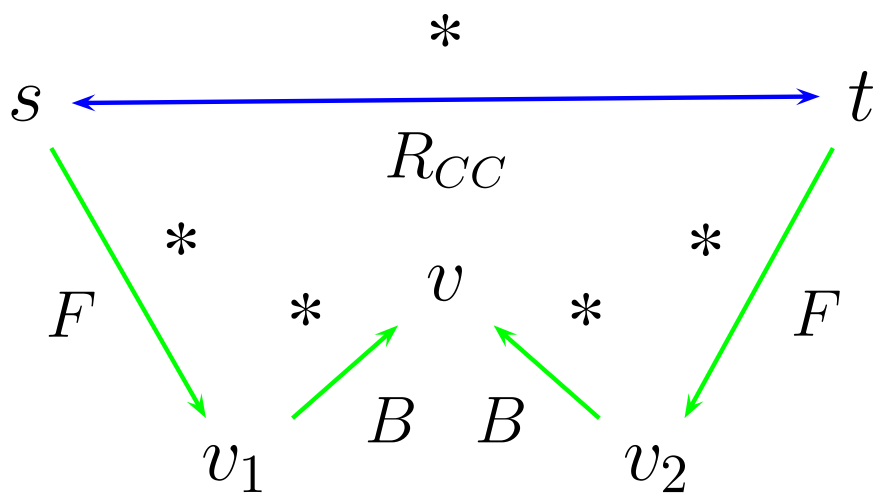
- Asymmetric completion gives flat terminating $F \cup B^{-}$.
- F not confluent, $\therefore$ non-deterministic search for v.

Finally

Recall we need to check the inclusion:

$$\leftrightarrow^*_{R_{CC}} \;\subseteq\; \rightarrow^*_{F \cup B} \circ \leftarrow^*_{F \cup B}$$

Now the picture looks like this:



Do we need to check this for all pair of terms s, t?

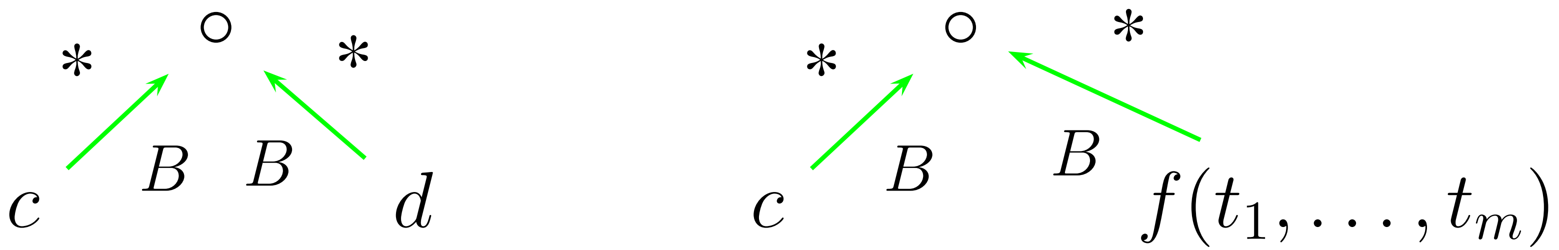
Towards the Main Theorem

If (s, t) is a counter-example and

$$s \longrightarrow_F^+ v_1, \qquad t \longrightarrow_F^+ v_2$$

then (v_1, v_2) is a smaller counter-example.
$\therefore$ we only test for F-irreducible terms s, t.

- If at least one of s or t is a constant, we need to check if



but B-rules are of a special form.

Towards the Main Theorem

If both s and t are not constants, and

$$s \rightarrow^{*}_{R_{CC}} u \leftarrow^{*}_{R_{CC}} t$$

then
- If u is not a constant, then there is a smaller counter-example in the arguments of s and t.

$$s = f(s_1, \ldots, s_m) \qquad\qquad t = f(t_1, \ldots, t_m)$$

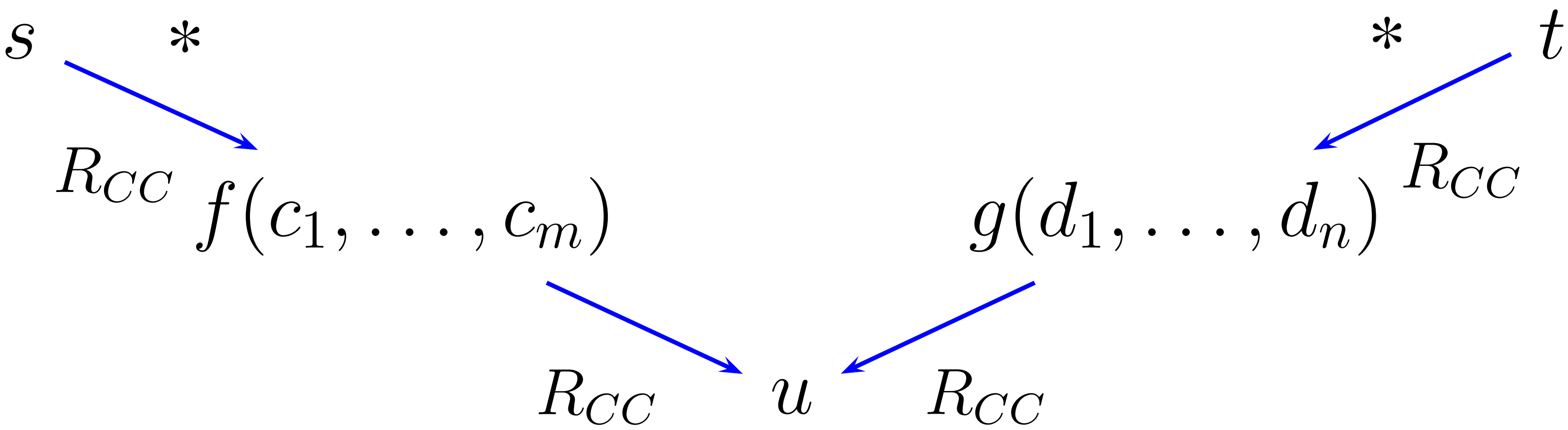
$$R_{CC} \downarrow * \qquad\qquad * \downarrow R_{CC}$$

$$f(\_, \ldots, \_) \qquad\qquad f(\_, \ldots, \_)$$

$$\searrow_{R_{CC}} \quad f(c_1, \ldots, c_m) \quad \swarrow_{R_{CC}}$$

Towards the Main Theorem

- If u is a constant, then



If either $f \neq g$ or some c_i and d_i are not equivalent (modulo R_{CC}), then (s, t) is a witness to non-confluence.

Technical Theorem

IRRSIG(c) : { $f c_1 \ldots c_n$: c is equivalent to this,
$f c_1 \ldots c_n$ represents an F-irreducible term}

IRRCON(c) : { d: d is equivalent to c,
d is F-irreducible}

Theorem. $\mathbb{R}$ is confluent iff

- $IRRSIG(c)$ contains at most one element
- if $IRRSIG(c) = \{f(c_1 \ldots c_n)\}$, then for all $c' \in IRRCON(c)$, there is a rule $c' \rightarrow f(c'_1 \ldots c'_n)$ in B s.t. c_i and c'_i are equivalent under R_{CC}
- for all $d, e \in IRRCON(c)$, $d \rightarrow^*_B \circ \leftarrow^*_B e$.

Complete Algorithm

Input: Finite set of ground rewrite rules $\mathbb{R}$

- Flatten $\mathbb{R}$ to R
- Construct abstract congruence closure R_{CC} for R
- Construct abstract rewrite closure (F, B) for R
- Compute $IRRSIG(c)$ and $IRRCON(c)$
- Return confluent if all the three conditions of the main theorem hold, not confluent otherwise

Complexity:

- Each step can be carried out in polynomial time.
- Rewrite closure complexity is exponential in the maximum arity of the function symbols in Σ
- But wlog the maximum arity can be bounded by 2

Back to Example

Recall

$$\mathbb{R}_0 \;=\; \{a \to fab,\ fab \to fba\}$$

Abstract rewrite closure represents the rewrite relation induced by $\mathbb{R}_0$:

$$\begin{aligned} E &= \{a \to c_0,\ b \to c_1,\ fc_0c_1 \to c_2\} \\ F &= E \cup \{c_0 \to c_2\} \\ B &= E^{-} \cup \{c_i \to fc_1c_0,\ c_i \to fc_1c_2,\ i = i, 2\} \end{aligned}$$

Abstract congruence closure represents the congruence relation induced by $\mathbb{R}_0$:

$$R_{CC} = \{\{a, fc_2c_1, fc_1c_2, c_0\} \to c_2,\ b \to c_1,\};$$

Back to Example 2

$$
\begin{aligned}
\mathbb{R}_0 &= \{a \longrightarrow fab,\ fab \longrightarrow fba\} \\
E &= \{a \longrightarrow c_0,\ b \longrightarrow c_1,\ fc_0c_1 \longrightarrow c_2\} \\
F &= E \cup \{c_0 \longrightarrow c_2\} \\
B &= E^{-} \cup \{c_i \longrightarrow fc_1c_0,\ c_i \longrightarrow fc_1c_2,\ i = i, 2\} \\
R_{CC} &= \{\{a, fc_2c_1, fc_1c_2, c_0\} \longrightarrow c_2,\ b \longrightarrow c_1, \};
\end{aligned}
$$

Now:

$$
\begin{aligned}
IRRSIG(c_1) &= \emptyset \\
IRRSIG(c_2) &= \{fc_1c_2, fc_2c_1\}
\end{aligned}
$$

Oh!, we fail right here. $\therefore$, $\mathbb{R}_0$ is not confluent.

Summary

- Confluence of Ground TRS has a polynomial time decision procedure.
- Using the same techniques, the result generalizes to TRS $\mathbb{R}$ s.t. for every rule $s \to t \in \mathbb{R}$:
  - Variables occur at depth at most one in s and t
  - No variable is repeated in $s \to t$
  - Exponential in the arity—can't be avoided
- The main underlying concept is that of top-stabilizability of a equivalence class c:

$$s \to^*_{R_{CC}} f(c_1, \ldots, c_n) \to_{R_{CC}} c$$

and s is F-irreducible.

Shallow Linear Systems

- Shallow: Do not need to use “abstraction”
- Linear: Do not need to consider variable overlaps for constructing rewrite closure

Future Work

- New decidability results for confluence of other classes of rewrite systems?
- New complexity results?
- Logical descriptions of other decision procedures?

References:

- Abstract congruence closure [BachmairTiwariVigneron:JAR2002]
- Abstract rewrite closures [Tiwari:FSTTCS2001]