Automatically Finding Non-CR Examples in Term Rewriting

Hans Zantema

Technische Universiteit Eindhoven and Radboud Universiteit Nijmegen

IWC, TU Eindhoven, June 28, 2013

Our goal:

Our goal:

Given a set of abstract rewrite properties, automatically find a TRS satisfying these properties

Our goal:

Given a set of abstract rewrite properties, automatically find a TRS satisfying these properties
(preferrably by exploiting the power of SAT / SMT / constraint solving)

Our goal:

Given a set of abstract rewrite properties, automatically find a TRS satisfying these properties
(preferrably by exploiting the power of SAT / SMT / constraint solving)

Last year at IWC in Nagoya we did this for finite ARSs, yielding some remarkable examples for which finding them by hand would be a hard job

Our goal:

Given a set of abstract rewrite properties, automatically find a TRS satisfying these properties
(preferrably by exploiting the power of SAT / SMT / constraint solving)

Last year at IWC in Nagoya we did this for finite ARSs, yielding some remarkable examples for which finding them by hand would be a hard job

However, for many sets of properties no finite ARS exists, but a TRS exists

Our goal:

Given a set of abstract rewrite properties, automatically find a TRS satisfying these properties
(preferrably by exploiting the power of SAT / SMT / constraint solving)

Last year at IWC in Nagoya we did this for finite ARSs, yielding some remarkable examples for which finding them by hand would be a hard job

However, for many sets of properties no finite ARS exists, but a TRS exists

For instance, the single rule $f(a) \to a$ is terminating and its reverse $a \to f(a)$ is non-terminating, while no finite ARS has this combination of properties

Leading example

Find a TRS that is locally confluent (WCR), but not confluent (CR), and for which the reverse is terminating (SN)

Leading example

Find a TRS that is locally confluent (WCR), but not confluent (CR), and for which the reverse is terminating (SN)

Solution:

$$
\begin{array}{rcl}
a & \rightarrow & b \\
a & \rightarrow & f(a) \\
b & \rightarrow & f(f(b))
\end{array}
$$

Leading example

Find a TRS that is locally confluent (WCR), but not confluent (CR), and for which the reverse is terminating (SN)

Solution:

$$\begin{array}{lcl} a & \to & b \\ a & \to & f(a) \\ b & \to & f(f(b)) \end{array}$$

$$\begin{tikzcd}
b \arrow[dd] & a \arrow[l] \arrow[d] & \\
 & f(a) \arrow[r] \arrow[d] & f(b) \arrow[dd] \\
f^2(b) \arrow[dd] & f^2(a) \arrow[l] \arrow[d] & \\
 & f^3(a) \arrow[r] \arrow[d] & f^3(b) \arrow[dd] \\
f^4(b) \arrow[dd] & f^4(a) \arrow[l] \arrow[d] & \\
 & f^5(a) \arrow[r] \arrow[d] & f^5(b) \arrow[d] \\
\ldots & \ldots & \ldots
\end{tikzcd}$$

Our goal: find such an example fully automatically

Our goal: find such an example fully automatically

In which space of TRSs are we looking?

Our goal: find such an example fully automatically

In which space of TRSs are we looking?

- Set of finite ARSs (= TRSs in which every lhs and rhs is a constant)?

Our goal: find such an example fully automatically

In which space of TRSs are we looking?

- Set of finite ARSs (= TRSs in which every lhs and rhs is a constant)?

  Too restrictive: if reverse of R is terminating then R is terminating too, and no such example exists by Newman's Lemma

Our goal: find such an example fully automatically

In which space of TRSs are we looking?

- Set of finite ARSs (= TRSs in which every lhs and rhs is a constant)?

  Too restrictive: if reverse of R is terminating then R is terminating too, and no such example exists by Newman's Lemma

- Set of all TRSs?

Our goal: find such an example fully automatically

In which space of TRSs are we looking?

- Set of finite ARSs ($=$ TRSs in which every lhs and rhs is a constant)?

  Too restrictive: if reverse of R is terminating then R is terminating too, and no such example exists by Newman's Lemma

- Set of all TRSs?

  Far too large

Our goal: find such an example fully automatically

In which space of TRSs are we looking?

- Set of finite ARSs (= TRSs in which every lhs and rhs is a constant)?

  Too restrictive: if reverse of R is terminating then R is terminating too, and no such example exists by Newman's Lemma

- Set of all TRSs?

  Far too large

We choose the smallest class of TRSs strictly including finite ARSs and allowing terminating TRSs for which the reverse is not terminating:

Our goal: find such an example fully automatically

In which space of TRSs are we looking?

- Set of finite ARSs ($=$ TRSs in which every lhs and rhs is a constant)?

  Too restrictive: if reverse of R is terminating then R is terminating too, and no such example exists by Newman's Lemma

- Set of all TRSs?

  Far too large

We choose the smallest class of TRSs strictly including finite ARSs and allowing terminating TRSs for which the reverse is not terminating:

Ground TRSs over a single unary symbol f and a finite set of constants

Ground TRSs over a single unary symbol f and a finite set of constants

Ground TRSs over a single unary symbol f and a finite set of constants

So, we only allow rules of the shape

$$f^n(a) \rightarrow f^k(b)$$

Ground TRSs over a single unary symbol f and a finite set of constants

So, we only allow rules of the shape

$$f^n(a) \to f^k(b)$$

By adding extra constants all such rules can be mimicked by a combination of rules of the shape

$$a \to b, \;\; f(a) \to b, \;\; a \to f(b)$$

Ground TRSs over a single unary symbol f and a finite set of constants

So, we only allow rules of the shape

$$f^n(a) \to f^k(b)$$

By adding extra constants all such rules can be mimicked by a combination of rules of the shape

$$a \to b, \quad f(a) \to b, \quad a \to f(b)$$

Fixing a number n of constants, let T_1 be the TRS of all $3n^2$ such rules

Ground TRSs over a single unary symbol f and a finite set of constants

So, we only allow rules of the shape

$$f^n(a) \to f^k(b)$$

By adding extra constants all such rules can be mimicked by a combination of rules of the shape

$$a \to b, \;\; f(a) \to b, \;\; a \to f(b)$$

Fixing a number n of constants, let T_1 be the TRS of all $3n^2$ such rules

Our search space will consist of the subTRSs of T_1

In order to express composition we also consider T_2 being the TRS of all $6n^2$ rules of the shape

$$a \to b, f(a) \to b, a \to f(b), a \to f(f(b)), f(f(a)) \to b, f(a) \to f(b)$$

In order to express composition we also consider T_2 being the TRS of all $6n^2$ rules of the shape

$$a \to b, f(a) \to b, a \to f(b), a \to f(f(b)), f(f(a)) \to b, f(a) \to f(b)$$

For instance, if $a \to f(b) \in R$ and $b \to f(c) \in S$, then $a \to_{R \cdot S} f(f(c))$

In order to express composition we also consider T_2 being the TRS of all $6n^2$ rules of the shape

$$a \to b, f(a) \to b, a \to f(b), a \to f(f(b)), f(f(a)) \to b, f(a) \to f(b)$$

For instance, if $a \to f(b) \in R$ and $b \to f(c) \in S$, then $a \to_{R \cdot S} f(f(c))$

Let $\mathrm{comp}(R, S)$ be the subTRS of T_2 mimicking all such compositions, so in this case $a \to f(f(c)) \in \mathrm{comp}(R, S)$

In order to express composition we also consider T_2 being the TRS of all $6n^2$ rules of the shape

$$a \to b, f(a) \to b, a \to f(b), a \to f(f(b)), f(f(a)) \to b, f(a) \to f(b)$$

For instance, if $a \to f(b) \in R$ and $b \to f(c) \in S$, then $a \to_{R \cdot S} f(f(c))$

Let $\mathrm{comp}(R, S)$ be the subTRS of T_2 mimicking all such compositions, so in this case $a \to f(f(c)) \in \mathrm{comp}(R, S)$

Theorem

1. *If* $R, S \subseteq T_1$ *then* $\to_{\mathrm{comp}(R,S)} = \to_R \cdot \to_S$.
2. *If* $R, S \subseteq T_2$ *then* $\to_{\mathrm{comp}(R,S)} \subseteq \to_R \cdot \to_S$.

Local confluence

Local confluence

Define inverse and reflexive closure:

$$\mathrm{inv}(R) = \{r \to \ell \mid \ell \to r \in R\}$$

$$\mathrm{rc}(R) = R \cup \{a \to a \mid a \in A\}$$

Local confluence

Define inverse and reflexive closure:

$$\mathrm{inv}(R) = \{r \to \ell \mid \ell \to r \in R\}$$

$$\mathrm{rc}(R) = R \cup \{a \to a \mid a \in A\}$$

Theorem

Let $R \subseteq T_1$, $R_1 = \mathrm{rc}(R)$, $R_{i+1} = \mathrm{comp}(R_i, R_i)$ *for* $i > 0$, *and*

$$\mathrm{comp}(\mathrm{inv}(R), R) \;\subseteq\; \mathrm{comp}(R_i, \mathrm{inv}(R_i))$$

for some $i > 0$
Then $WCR(R)$

Local confluence

Define inverse and reflexive closure:

$$\mathrm{inv}(R) = \{r \to \ell \mid \ell \to r \in R\}$$

$$\mathrm{rc}(R) = R \cup \{a \to a \mid a \in A\}$$

Theorem

Let $R \subseteq T_1$, $R_1 = \mathrm{rc}(R)$, $R_{i+1} = \mathrm{comp}(R_i, R_i)$ *for* $i > 0$, *and*

$$\mathrm{comp}(\mathrm{inv}(R), R) \ \subseteq \ \mathrm{comp}(R_i, \mathrm{inv}(R_i))$$

for some $i > 0$
Then $WCR(R)$

If WCR is required, we express this by the slightly stronger requirement $\mathrm{comp}(\mathrm{inv}(R), R) \ \subseteq \ \mathrm{comp}(R_i, \mathrm{inv}(R_i))$ for $i = 2$ or $i = 3$

Projection to finite ARS

Projection to finite ARS

Rough idea: identify $f(f(x))$ with x

Projection to finite ARS

Rough idea: identify $f(f(x))$ with x

More precisely: for every constant a define new constants a_0, a_1

$$\pi(a \to b) = \{a_0 \to b_0, a_1 \to b_1\}$$

$$\pi(a \to f(b)) = \{a_0 \to b_1, a_1 \to b_0\}$$

$$\pi(f(a) \to b) = \{a_0 \to b_1, a_1 \to b_0\}$$

Projection to finite ARS

Rough idea: identify $f(f(x))$ with x

More precisely: for every constant a define new constants a_0, a_1

$$\pi(a \to b) = \{a_0 \to b_0, a_1 \to b_1\}$$

$$\pi(a \to f(b)) = \{a_0 \to b_1, a_1 \to b_0\}$$

$$\pi(f(a) \to b) = \{a_0 \to b_1, a_1 \to b_0\}$$

Theorem

If $CR(R)$, then $CR(\pi(R))$

Projection to finite ARS

Rough idea: identify $f(f(x))$ with x

More precisely: for every constant a define new constants a_0, a_1

$$\pi(a \to b) = \{a_0 \to b_0, a_1 \to b_1\}$$

$$\pi(a \to f(b)) = \{a_0 \to b_1, a_1 \to b_0\}$$

$$\pi(f(a) \to b) = \{a_0 \to b_1, a_1 \to b_0\}$$

Theorem

If $CR(R)$, then $CR(\pi(R))$

If $\neg\mathrm{CR}(R)$ is required, we express this by the stronger requirement $\neg\mathrm{CR}(\pi(R))$, which is about finite ARS, so can be expressed by earlier techniques

Termination

Where for WCR and $\neg$CR we only found approximations, termination can be expressed exactly

Termination

Where for WCR and $\neg$CR we only found approximations, termination can be expressed exactly

Theorem

A ground TRS R over $\{f\} \cup A$ is terminating if and only if a map $W : A \to \mathbf{R}$ exists such that $W(a) + n > W(b) + k$ for every $f^n(a) \to f^k(b) \in R$

Where for WCR and $\neg$CR we only found approximations, termination can be expressed exactly

Theorem

A ground TRS R over $\{f\} \cup A$ is terminating if and only if a map $W : A \to \mathbf{R}$ exists such that $W(a) + n > W(b) + k$ for every $f^n(a) \to f^k(b) \in R$

This criterium for SN is easily expressed in SMT, satisfiability modulo theory of linear inequalities, where until now everything was in propositional SAT

Implementation

Implementation

For finite ARSs we already developed a tool CARPA: Counter examples of Abstract Rewriting Produced Automatically

Implementation

For finite ARSs we already developed a tool CARPA: Counter examples of Abstract Rewriting Produced Automatically

We modified the CARPA input language to deal with the new TRS features

Implementation

For finite ARSs we already developed a tool CARPA: Counter examples of Abstract Rewriting Produced Automatically

We modified the CARPA input language to deal with the new TRS features

Example

`x1=inv(1)`	$x1$ is inverse of basic TRS 1
`sn(x1)`	$x1$ is terminating

Implementation

For finite ARSs we already developed a tool CARPA: Counter examples of Abstract Rewriting Produced Automatically

We modified the CARPA input language to deal with the new TRS features

Example

`x1=inv(1)`	$x1$ is inverse of basic TRS 1
`sn(x1)`	$x1$ is terminating
`x2=peak(1,1)`	
`x3=rc(1)`	
`x3=comp(x3,x3)`	
`x3=val(x3,x3)`	describes WCR(1) as discussed:
`subs(x2,x3)`	peak contained in valley

For finite ARSs we already developed a tool CARPA: Counter examples of Abstract Rewriting Produced Automatically

We modified the CARPA input language to deal with the new TRS features

Example

`x1=inv(1)`	$x1$ is inverse of basic TRS 1
`sn(x1)`	$x1$ is terminating
`x2=peak(1,1)`	
`x3=rc(1)`	
`x3=comp(x3,x3)`	
`x3=val(x3,x3)`	describes WCR(1) as discussed:
`subs(x2,x3)`	peak contained in valley
`x1=mod2(1)`	$x1$ is projection to finite ARS of 1
`ncr(x1)`	non-confluent by earlier techniques

Applying our tool to this input, and specify $\#A = 3$

Applying our tool to this input, and specify $\#A = 3$

- generates a formula expressing all these ingredients

Applying our tool to this input, and specify $\#A = 3$

- generates a formula expressing all these ingredients
- calls the SMT solver YICES

Applying our tool to this input, and specify $\#A = 3$

- generates a formula expressing all these ingredients
- calls the SMT solver YICES
- inspects the solution found by YICES, and transforms this to the following output

```
1 -> 2
1 -> f(1)
2 -> f(3)
3 -> f(2)
```

Applying our tool to this input, and specify $\#A = 3$

- generates a formula expressing all these ingredients
- calls the SMT solver YICES
- inspects the solution found by YICES, and transforms this to the following output

```
1 -> 2
1 -> f(1)
2 -> f(3)
3 -> f(2)
```

which is indeed a TRS satisfying the given requirements

Conclusions

Conclusions

- We developed a method for automatically finding TRSs having a given set of properties, in particular WCR($\to$), $\neg$CR($\to$), SN($\leftarrow$)

Conclusions

- We developed a method for automatically finding TRSs having a given set of properties, in particular $\mathrm{WCR}(\rightarrow)$, $\neg\mathrm{CR}(\rightarrow)$, $\mathrm{SN}(\leftarrow)$
- The real work is done by an SMT solver

Conclusions

- We developed a method for automatically finding TRSs having a given set of properties, in particular WCR($\rightarrow$), $\neg$CR($\rightarrow$), SN($\leftarrow$)
- The real work is done by an SMT solver
- We restricted to ground TRSs over constants and one single unary symbol: both restrictive and a substantial extension compared to finite ARSs

Conclusions

- We developed a method for automatically finding TRSs having a given set of properties, in particular WCR($\to$), $\neg$CR($\to$), SN($\leftarrow$)
- The real work is done by an SMT solver
- We restricted to ground TRSs over constants and one single unary symbol: both restrictive and a substantial extension compared to finite ARSs
- If $\to^*$ comes in: arbitrary number of steps, then we approximate, or project to a finite ARS, loosing completeness

Conclusions

- We developed a method for automatically finding TRSs having a given set of properties, in particular WCR($\rightarrow$), $\neg$CR($\rightarrow$), SN($\leftarrow$)
- The real work is done by an SMT solver
- We restricted to ground TRSs over constants and one single unary symbol: both restrictive and a substantial extension compared to finite ARSs
- If $\rightarrow^*$ comes in: arbitrary number of steps, then we approximate, or project to a finite ARS, loosing completeness
- For this restricted class WCR and CR are decidable; we believe encoding corresponding algorithms in SAT/SMT will not better serve our goal

Conclusions

- We developed a method for automatically finding TRSs having a given set of properties, in particular WCR($\rightarrow$), $\neg$CR($\rightarrow$), SN($\leftarrow$)
- The real work is done by an SMT solver
- We restricted to ground TRSs over constants and one single unary symbol: both restrictive and a substantial extension compared to finite ARSs
- If $\rightarrow^*$ comes in: arbitrary number of steps, then we approximate, or project to a finite ARS, loosing completeness
- For this restricted class WCR and CR are decidable; we believe encoding corresponding algorithms in SAT/SMT will not better serve our goal
- Termination is expressed exactly