Decision Procedures for Loop Detection

René Thiemann

LuFG Informatik 2, RWTH Aachen
Institute of Computer Science, University of Innsbruck

Dagstuhl Seminar on Deduction and Decision Procedures

Joint work with J. Giesl and P. Schneider-Kamp

Outline

Main motivation: Non-termination analysis to detect bugs in programs

- Programs can be transformed into TRSs

$\Rightarrow$ Benefit from termination analysis of TRSs

Disproving termination of TRSs

- Use dependency pairs and narrowing for loop detection [FroCoS '05]

Outline

Main motivation: Non-termination analysis to detect bugs in programs

- Programs can be transformed into TRSs

⇒ Benefit from termination analysis of TRSs

Disproving termination of TRSs

- Use dependency pairs and narrowing for loop detection [FroCoS '05]

Programs often use innermost evaluation

⇒ Disproving innermost termination of TRSs

- Decision procedure to detect innermost loops [New unpublished work]

Example (ACL2-function computing factorial function)

$(\mathit{defun}$ factorial (x) (fact 0 $x))$

$(\mathit{defun}$ fact $(x\ y)$ (if $(==\ x\ y)$
  1
  $(\times$ (fact $(+$ 1 $x)$ $y)$ $(+$ 1 $x))))$

Example (ACL2-function computing factorial function)

$$(\mathit{defun}\ \mathsf{factorial}\ (x)\ (\mathsf{fact}\ 0\ x))$$

$$\begin{array}{l}(\mathit{defun}\ \mathsf{fact}\ (x\ y)\ (\mathsf{if}\ (==\ x\ y) \\ \quad 1 \\ \quad (\times\ (\mathsf{fact}\ (+\ 1\ x)\ y)\ (+\ 1\ x))))\end{array}$$

Example (TRS computing factorial function)

$$\begin{array}{rcl}\mathsf{factorial}(x) & \to & \mathsf{fact}(0, x) \\ \mathsf{fact}(x, y) & \to & \mathsf{if}(x == y, x, y) \\ \mathsf{if}(\mathsf{true}, x, y) & \to & \mathsf{s}(0) \\ \mathsf{if}(\mathsf{false}, x, y) & \to & \mathsf{fact}(\mathsf{s}(x), y) \times \mathsf{s}(x)\end{array}$$

Example (TRS computing factorial function)

$$
\begin{aligned}
\mathsf{factorial}(x) &\to \mathsf{fact}(0, x)\\
\mathsf{fact}(x, y) &\to \mathsf{if}(x == y, x, y)\\
\mathsf{if}(\mathsf{true}, x, y) &\to \mathsf{s}(0)\\
\mathsf{if}(\mathsf{false}, x, y) &\to \mathsf{fact}(\mathsf{s}(x), y) \times \mathsf{s}(x)\\
0 + y &\to y\\
\mathsf{s}(x) + y &\to \mathsf{s}(x + y)\\
0 \times y &\to 0\\
\mathsf{s}(x) \times y &\to y + (x \times y)\\
x == y &\to \mathsf{chk}(\mathsf{eq}(x, y))\\
\mathsf{eq}(x, x) &\to \mathsf{true}\\
\mathsf{chk}(\mathsf{true}) &\to \mathsf{true}\\
\mathsf{chk}(\mathsf{eq}(x, y)) &\to \mathsf{false}
\end{aligned}
$$

- Encoding of built-in equality of ACL2 [Vroon '07]
- Only works for innermost termination!

$$0 == 0 \to_{\mathcal{R}} \mathsf{chk}(\mathsf{eq}(0, 0)) \to_{\mathcal{R}} \mathsf{false}$$

Example (TRS computing factorial function)

$$
\begin{array}{rcl}
\mathsf{factorial}(x) & \to & \mathsf{fact}(\mathsf{0}, x) \\
\mathsf{fact}(x, y) & \to & \mathsf{if}(x == y, x, y) \\
\mathsf{if}(\mathsf{true}, x, y) & \to & \mathsf{s}(\mathsf{0}) \\
\mathsf{if}(\mathsf{false}, x, y) & \to & \mathsf{fact}(\mathsf{s}(x), y) \times \mathsf{s}(x) \\
\mathsf{0} + y & \to & y \\
\mathsf{s}(x) + y & \to & \mathsf{s}(x + y) \\
\mathsf{0} \times y & \to & \mathsf{0} \\
\mathsf{s}(x) \times y & \to & y + (x \times y) \\
x == y & \to & \mathsf{chk}(\mathsf{eq}(x, y)) \\
\mathsf{eq}(x, x) & \to & \mathsf{true} \\
\mathsf{chk}(\mathsf{true}) & \to & \mathsf{true} \\
\mathsf{chk}(\mathsf{eq}(x, y)) & \to & \mathsf{false}
\end{array}
$$

- Encoding of built-in equality of ACL2 [Vroon '07]
- Only works for innermost termination!

$\Rightarrow$ Result that innermost-termination = termination for non-overlapping TRSs useless here

Example (TRS computing factorial function)

$$
\begin{array}{rcl}
\mathsf{factorial}(x) & \to & \mathsf{fact}(0, x) \\
\mathsf{fact}(x, y) & \to & \mathsf{if}(x == y, x, y) \\
\mathsf{if}(\mathsf{true}, x, y) & \to & \mathsf{s}(0) \\
\mathsf{if}(\mathsf{false}, x, y) & \to & \mathsf{fact}(\mathsf{s}(x), y) \times \mathsf{s}(x) \\
0 + y & \to & y \\
\mathsf{s}(x) + y & \to & \mathsf{s}(x + y) \\
0 \times y & \to & 0 \\
\mathsf{s}(x) \times y & \to & y + (x \times y) \\
x == y & \to & \mathsf{chk}(\mathsf{eq}(x, y)) \\
\mathsf{eq}(x, x) & \to & \mathsf{true} \\
\mathsf{chk}(\mathsf{true}) & \to & \mathsf{true} \\
\mathsf{chk}(\mathsf{eq}(x, y)) & \to & \mathsf{false}
\end{array}
$$

- A TRS $\mathcal{R}$ is non-terminating iff there is an infinite reduction

$$t_1 \to_{\mathcal{R}} t_2 \to_{\mathcal{R}} t_3 \to_{\mathcal{R}} \dots$$

- Often the cause for non-termination: loops

Example (TRS computing factorial function)

$$
\begin{array}{rcl}
\mathsf{factorial}(x) & \to & \mathsf{fact}(0, x) \\
\mathsf{fact}(x, y) & \to & \mathsf{if}(x == y, x, y) \\
\mathsf{if}(\mathsf{true}, x, y) & \to & \mathsf{s}(0) \\
\mathsf{if}(\mathsf{false}, x, y) & \to & \mathsf{fact}(\mathsf{s}(x), y) \times \mathsf{s}(x) \\
0 + y & \to & y \\
\mathsf{s}(x) + y & \to & \mathsf{s}(x + y) \\
0 \times y & \to & 0 \\
\mathsf{s}(x) \times y & \to & y + (x \times y) \\
x == y & \to & \mathsf{chk}(\mathsf{eq}(x, y)) \\
\mathsf{eq}(x, x) & \to & \mathsf{true} \\
\mathsf{chk}(\mathsf{true}) & \to & \mathsf{true} \\
\mathsf{chk}(\mathsf{eq}(x, y)) & \to & \mathsf{false}
\end{array}
$$

$$
\begin{array}{lllll}
\underline{\mathsf{fact}(x, y)} & \to_{\mathcal{R}} & \mathsf{if}(\underline{x == y}, x, y) & \to_{\mathcal{R}} & \mathsf{if}(\underline{\mathsf{chk}(\mathsf{eq}(x, y))}, x, y) \\
 & \to_{\mathcal{R}} & \underline{\mathsf{if}(\mathsf{false}, x, y)} & \to_{\mathcal{R}} & \mathsf{fact}(\mathsf{s}(x), y) \times \mathsf{s}(x)
\end{array}
$$

Definition (Looping TRS)

$\mathcal{R}$ is looping iff there are t, C, and μ such that $t \to_{\mathcal{R}}^{+} C[t\mu]$.

Definition (Looping TRS)

$\mathcal{R}$ is looping iff there are t, C, and μ such that $t \rightarrow^+_{\mathcal{R}} C[t\mu]$.

Infinite reduction: $t \rightarrow^+_{\mathcal{R}} C[t\mu] \rightarrow^+_{\mathcal{R}} C[C\mu[t\mu^2]] \rightarrow^+_{\mathcal{R}} C[C\mu[C\mu^2[t\mu^3]]] \rightarrow^+_{\mathcal{R}}$

Definition (Looping TRS)

$\mathcal{R}$ is looping iff there are t, C, and μ such that $t \to_{\mathcal{R}}^{+} C[t\mu]$.

Problem

How to find t, C, μ, and the reduction?

- Restriction of t possible [FroCoS '05]:
  if $\mathcal{R}$ is looping then $t \to_{\mathcal{R}}^{+} C[t\mu]$ where $t = \ell\sigma$ for some $\ell \to r \in \mathcal{R}$

  But: one still has to consider all rules for the reduction.

- better: use dependency pairs
  $\Rightarrow$ search space for loops is decreased significantly

Definition (Looping TRS)

$\mathcal{R}$ is looping iff there are t, C, and μ such that $t \to_{\mathcal{R}}^{+} C[t\mu]$.

Problem

How to find t, C, μ, and the reduction?

- Restriction of t possible [FroCoS '05]:
  if $\mathcal{R}$ is looping then $t \to_{\mathcal{R}}^{+} C[t\mu]$ where $t = \ell\sigma$ for some $\ell \to r \in \mathcal{R}$

  But: one still has to consider all rules for the reduction.

- better: use dependency pairs
  $\Rightarrow$ search space for loops is decreased significantly

Definition (Looping TRS)

$\mathcal{R}$ is looping iff there are t, C, and μ such that $t \to_{\mathcal{R}}^{+} C[t\mu]$.

Problem

How to find t, C, μ, and the reduction?

- Restriction of t possible [FroCoS '05]:
  if $\mathcal{R}$ is looping then $t \to_{\mathcal{R}}^{+} C[t\mu]$ where $t = \ell\sigma$ for some $\ell \to r \in \mathcal{R}$

  But: one still has to consider all rules for the reduction.

- better: use dependency pairs
  $\Rightarrow$ search space for loops is decreased significantly

Definition (Looping TRS)

$\mathcal{R}$ is looping iff there are t, C, and μ such that $t \to_{\mathcal{R}}^{+} C[t\mu]$.

Problem

How to find t, C, μ, and the reduction?

- Restriction of t possible [FroCoS '05]:
  if $\mathcal{R}$ is looping then $t \to_{\mathcal{R}}^{+} C[t\mu]$ where $t = \ell\sigma$ for some $\ell \to r \in \mathcal{R}$

  But: one still has to consider all rules for the reduction.

- better: use dependency pairs
  $\Rightarrow$ search space for loops is decreased significantly

Problem

What about loopingness for innermost rewriting?

Example (Build dependency pairs $DP(\mathcal{R}) \sim$ (recursive) calls)

$\mathcal{R}$	$DP(\mathcal{R})$
$\mathsf{factorial}(x) \to \mathsf{fact}(0, x)$	$\mathsf{factorial}^\sharp(x) \to \mathsf{fact}^\sharp(0, x)$
$\mathsf{fact}(x, y) \to \mathsf{if}(x == y, x, y)$	$\mathsf{fact}^\sharp(x, y) \to \mathsf{if}^\sharp(x == y, x, y)$
	$\mathsf{fact}^\sharp(x, y) \to x ==^\sharp y$
$\mathsf{if}(\mathsf{true}, x, y) \to \mathsf{s}(0)$	
$\mathsf{if}(\mathsf{false}, x, y) \to \mathsf{fact}(\mathsf{s}(x), y) \times \mathsf{s}(x)$	$\mathsf{if}^\sharp(\mathsf{false}, x, y) \to \mathsf{fact}(\mathsf{s}(x), y) \times^\sharp \mathsf{s}(x)$
	$\mathsf{if}^\sharp(\mathsf{false}, x, y) \to \mathsf{fact}^\sharp(\mathsf{s}(x), y)$
$0 + y \to y$	
$\mathsf{s}(x) + y \to \mathsf{s}(x + y)$	$\mathsf{s}(x) +^\sharp y \to x +^\sharp y$
$0 \times y \to 0$	
$\mathsf{s}(x) \times y \to y + (x \times y)$	$\mathsf{s}(x) \times^\sharp y \to y +^\sharp (x \times y)$
	$\mathsf{s}(x) \times^\sharp y \to x \times^\sharp y$
$x == y \to \mathsf{chk}(\mathsf{eq}(x, y))$	$x ==^\sharp y \to \mathsf{chk}^\sharp(\mathsf{eq}(x, y))$
	$x ==^\sharp y \to \mathsf{eq}^\sharp(x, y)$
$\mathsf{eq}(x, x) \to \mathsf{true}$	
$\mathsf{chk}(\mathsf{true}) \to \mathsf{true}$	
$\mathsf{chk}(\mathsf{eq}(x, y)) \to \mathsf{false}$	

Definition (Infinite Chain $\sim$ infinite sequence of function calls)

Let $\mathcal{P}$ be a set of dependency pairs, let $\mathcal{R}$ be a TRS.

An infinite $(\mathcal{P}, \mathcal{R})$-chain is an infinite reduction

$$t_1 \to_{\mathcal{P}} t_2 \to^*_{\mathcal{R}} t_3 \to_{\mathcal{P}} t_4 \to^*_{\mathcal{R}} \dots$$

Theorem (Termination Criterion [ArtsGiesl '00])

$\mathcal{R}$ is non-terminating iff there is an infinite $(DP(\mathcal{R}), \mathcal{R})$-chain.

Definition (Infinite Chain $\sim$ infinite sequence of function calls)

Let $\mathcal{P}$ be a set of dependency pairs, let $\mathcal{R}$ be a TRS.

An infinite $(\mathcal{P}, \mathcal{R})$-chain is an infinite reduction

$$t_1 \to_{\mathcal{P}} t_2 \to_{\mathcal{R}}^* t_3 \to_{\mathcal{P}} t_4 \to_{\mathcal{R}}^* \ldots$$

Theorem (Termination Criterion [ArtsGiesl '00])

$\mathcal{R}$ is non-terminating iff there is an infinite $(DP(\mathcal{R}), \mathcal{R})$-chain.

Main idea

Investigate DP problems $(\mathcal{P}, \mathcal{R})$ instead of TRS $\mathcal{R}$ directly:

$(\mathcal{P}, \mathcal{R})$-chains instead of $\to_{\mathcal{R}}$

Example (Detecting loops becomes simpler by DPs?)

$\mathcal{R}$

$$
\begin{aligned}
\mathsf{factorial}(x) &\to \mathsf{fact}(0, x) \\
\mathsf{fact}(x, y) &\to \mathsf{if}(x == y, x, y) \\
\\
\mathsf{if}(\mathsf{true}, x, y) &\to \mathsf{s}(0) \\
\mathsf{if}(\mathsf{false}, x, y) &\to \mathsf{fact}(\mathsf{s}(x), y) \times \mathsf{s}(x) \\
\\
0 + y &\to y \\
\mathsf{s}(x) + y &\to \mathsf{s}(x + y) \\
0 \times y &\to 0 \\
\mathsf{s}(x) \times y &\to y + (x \times y) \\
\\
x == y &\to \mathsf{chk}(\mathsf{eq}(x, y)) \\
\\
\mathsf{eq}(x, x) &\to \mathsf{true} \\
\mathsf{chk}(\mathsf{true}) &\to \mathsf{true} \\
\mathsf{chk}(\mathsf{eq}(x, y)) &\to \mathsf{false}
\end{aligned}
$$

$DP(\mathcal{R})$

$$
\begin{aligned}
\mathsf{factorial}^\sharp(x) &\to \mathsf{fact}^\sharp(0, x) \\
\mathsf{fact}^\sharp(x, y) &\to \mathsf{if}^\sharp(x == y, x, y) \\
\mathsf{fact}^\sharp(x, y) &\to x ==^\sharp y \\
\\
\mathsf{if}^\sharp(\mathsf{false}, x, y) &\to \mathsf{fact}(\mathsf{s}(x), y) \times^\sharp \mathsf{s}(x) \\
\mathsf{if}^\sharp(\mathsf{false}, x, y) &\to \mathsf{fact}^\sharp(\mathsf{s}(x), y) \\
\\
\mathsf{s}(x) +^\sharp y &\to x +^\sharp y \\
\\
\mathsf{s}(x) \times^\sharp y &\to y +^\sharp (x \times y) \\
\mathsf{s}(x) \times^\sharp y &\to x \times^\sharp y \\
x ==^\sharp y &\to \mathsf{chk}^\sharp(\mathsf{eq}(x, y)) \\
x ==^\sharp y &\to \mathsf{eq}^\sharp(x, y)
\end{aligned}
$$

Example (Detecting loops becomes simpler by DPs? Yes!)

First advantage: DP techniques detect non-terminating part

$\mathcal{R}$	$DP(\mathcal{R})$
$\mathsf{factorial}(x) \to \mathsf{fact}(0, x)$	$\mathsf{factorial}^\sharp(x) \to \mathsf{fact}^\sharp(0, x)$
$\mathsf{fact}(x, y) \to \mathsf{if}(x == y, x, y)$	$\mathsf{fact}^\sharp(x, y) \to \mathsf{if}^\sharp(x == y, x, y)$
	$\mathsf{fact}^\sharp(x, y) \to x ==^\sharp y$
$\mathsf{if}(\mathsf{true}, x, y) \to \mathsf{s}(0)$	
$\mathsf{if}(\mathsf{false}, x, y) \to \mathsf{fact}(\mathsf{s}(x), y) \times \mathsf{s}(x)$	$\mathsf{if}^\sharp(\mathsf{false}, x, y) \to \mathsf{fact}(\mathsf{s}(x), y) \times^\sharp \mathsf{s}(x)$
	$\mathsf{if}^\sharp(\mathsf{false}, x, y) \to \mathsf{fact}^\sharp(\mathsf{s}(x), y)$
$0 + y \to y$	
$\mathsf{s}(x) + y \to \mathsf{s}(x + y)$	$\mathsf{s}(x) +^\sharp y \to x +^\sharp y$
$0 \times y \to 0$	
$\mathsf{s}(x) \times y \to y + (x \times y)$	$\mathsf{s}(x) \times^\sharp y \to y +^\sharp (x \times y)$
	$\mathsf{s}(x) \times^\sharp y \to x \times^\sharp y$
$x == y \to \mathsf{chk}(\mathsf{eq}(x, y))$	$x ==^\sharp y \to \mathsf{chk}^\sharp(\mathsf{eq}(x, y))$
	$x ==^\sharp y \to \mathsf{eq}^\sharp(x, y)$
$\mathsf{eq}(x, x) \to \mathsf{true}$	
$\mathsf{chk}(\mathsf{true}) \to \mathsf{true}$	
$\mathsf{chk}(\mathsf{eq}(x, y)) \to \mathsf{false}$	

Example (Detecting loops becomes simpler by DPs? Yes!)

First advantage: DP techniques detect non-terminating part

$$
\begin{array}{rcl}
\mathsf{fact}^\sharp(x,y) & \to & \mathsf{if}^\sharp(x == y, x, y) \\
\mathsf{if}^\sharp(\mathsf{false}, x, y) & \to & \mathsf{fact}^\sharp(\mathsf{s}(x), y) \\
\hline
x == y & \to & \mathsf{chk}(\mathsf{eq}(x,y)) \\
\mathsf{eq}(x,x) & \to & \mathsf{true} \\
\mathsf{chk}(\mathsf{true}) & \to & \mathsf{true} \\
\mathsf{chk}(\mathsf{eq}(x,y)) & \to & \mathsf{false}
\end{array}
$$

Example (Detecting loops becomes simpler by DPs? Yes!)

First advantage: DP techniques detect non-terminating part

$$
\begin{array}{rcl}
\mathsf{fact}^\sharp(x,y) & \to & \mathsf{if}^\sharp(x == y, x, y) \\
\mathsf{if}^\sharp(\mathsf{false}, x, y) & \to & \mathsf{fact}^\sharp(\mathsf{s}(x), y) \\
\hline
x == y & \to & \mathsf{chk}(\mathsf{eq}(x,y)) \\
\mathsf{eq}(x,x) & \to & \mathsf{true} \\
\mathsf{chk}(\mathsf{true}) & \to & \mathsf{true} \\
\mathsf{chk}(\mathsf{eq}(x,y)) & \to & \mathsf{false}
\end{array}
$$

Definition (Looping TRS)

$\mathcal{R}$ is looping iff there are t, C, and μ such that $t \to_{\mathcal{R}}^{+} C[t\mu]$.

Definition (Looping DP Problem)

$(\mathcal{P}, \mathcal{R})$ is looping iff there are t and μ such that $t \to_{\mathcal{P}} \circ \to_{\mathcal{P},\mathcal{R}}^{*} t\mu$.

Infinite chain: $t \to_{\mathcal{P}} \circ \to_{\mathcal{P},\mathcal{R}}^{*} t\mu \to_{\mathcal{P}} \circ \to_{\mathcal{P},\mathcal{R}}^{*} t\mu^2 \to_{\mathcal{P}} \circ \to_{\mathcal{P},\mathcal{R}}^{*} t\mu^3 \to_{\mathcal{P}} \circ$

Example (Detecting loops becomes simpler by DPs? Yes!)

First advantage: DP techniques detect non-terminating part

$$
\begin{array}{rcl}
\mathsf{fact}^\sharp(x,y) & \to & \mathsf{if}^\sharp(x == y, x, y) \\
\mathsf{if}^\sharp(\mathsf{false}, x, y) & \to & \mathsf{fact}^\sharp(\mathsf{s}(x), y) \\
\hline
x == y & \to & \mathsf{chk}(\mathsf{eq}(x,y)) \\
\mathsf{eq}(x,x) & \to & \mathsf{true} \\
\mathsf{chk}(\mathsf{true}) & \to & \mathsf{true} \\
\mathsf{chk}(\mathsf{eq}(x,y)) & \to & \mathsf{false}
\end{array}
$$

Definition (Looping TRS)

$\mathcal{R}$ is looping iff there are t, C, and μ such that $t \to_{\mathcal{R}}^{+} C[t\mu]$.

Definition (Looping DP Problem)

$(\mathcal{P}, \mathcal{R})$ is looping iff there are t and μ such that $t \to_{\mathcal{P}} \circ \to_{\mathcal{P},\mathcal{R}}^{*} t\mu$.

Second advantage: we do not need a context C for a loop!

Example (Detecting loops becomes simpler by DPs? Yes!)

First advantage: DP techniques detect non-terminating part

$$\begin{array}{rcl} \mathsf{fact}^\sharp(x,y) & \to & \mathsf{if}^\sharp(x == y, x, y) \\ \mathsf{if}^\sharp(\mathsf{false}, x, y) & \to & \mathsf{fact}^\sharp(\mathsf{s}(x), y) \\ \hline x == y & \to & \mathsf{chk}(\mathsf{eq}(x,y)) \\ \mathsf{eq}(x,x) & \to & \mathsf{true} \\ \mathsf{chk}(\mathsf{true}) & \to & \mathsf{true} \\ \mathsf{chk}(\mathsf{eq}(x,y)) & \to & \mathsf{false} \end{array}$$

Definition (Looping TRS)

$\mathcal{R}$ is looping iff there are t, C, and μ such that $t \to_{\mathcal{R}}^{+} C[t\mu]$.

Definition (Looping DP Problem)

$(\mathcal{P}, \mathcal{R})$ is looping iff there are t and μ such that $t \to_{\mathcal{P}} \circ \to_{\mathcal{P},\mathcal{R}}^{*} t\mu$.

Second advantage: we do not need a context C for a loop!

Theorem (Equivalence of Loopingness [FroCoS '05])

$\mathcal{R}$ is looping iff $(DP(\mathcal{R}), \mathcal{R})$ is looping.

Definition (Looping DP Problem)

$(\mathcal{P}, \mathcal{R})$ is looping iff there are t and μ such that $t \to_{\mathcal{P}} \circ \to^*_{\mathcal{P},\mathcal{R}} t\mu$.

Problem: How to detect loops?

- Forward closures [LankfordMusser '78]
- Forward narrowing and backward narrowing
- Overlap closures [GuttagKapurMusser '83]

Definition (Looping DP Problem)

$(\mathcal{P}, \mathcal{R})$ is looping iff there are t and μ such that $t \to_{\mathcal{P}} \circ \to^*_{\mathcal{P},\mathcal{R}} t\mu$.

Problem: How to detect loops?

- Forward closures [LankfordMusser '78]
- Forward narrowing and backward narrowing
- Overlap closures [GuttagKapurMusser '83]

Theorem (Loop detection by narrowing [FroCoS '05])

$(\mathcal{P}, \mathcal{R})$ is looping if there is some $\ell \to r \in \mathcal{P}$ such that

- $r \rightsquigarrow^*_{\mathcal{P},\mathcal{R},\sigma} \ell'$ $\qquad$ *(*$r\sigma \to^*_{\mathcal{P},\mathcal{R}} \ell'$*)*
- $\ell\sigma$ *matches* ℓ' $\qquad$ *(*$\ell\sigma\mu = \ell'$*)*
- *Loop:* $\ell\sigma \to_{\mathcal{P}} r\sigma \to^*_{\mathcal{P},\mathcal{R}} \ell' = \ell\sigma\mu$

Theorem (Loop detection by narrowing [FroCoS '05])

$(\mathcal{P}, \mathcal{R})$ is looping if there is some $\ell \to r \in \mathcal{P}$ such that

- $r \rightsquigarrow^*_{\mathcal{P},\mathcal{R},\sigma} \ell'$
- *$\ell\sigma$ matches ℓ'*

Example

$$\begin{array}{rcl} \mathsf{fact}^\sharp(x, y) & \to & \mathsf{if}^\sharp(x == y, x, y) \\ \mathsf{if}^\sharp(\mathsf{false}, x, y) & \to & \mathsf{fact}^\sharp(\mathsf{s}(x), y) \\ \hline x == y & \to & \mathsf{chk}(\mathsf{eq}(x, y)) \\ \mathsf{eq}(x, x) & \to & \mathsf{true} \\ \mathsf{chk}(\mathsf{true}) & \to & \mathsf{true} \\ \mathsf{chk}(\mathsf{eq}(x, y)) & \to & \mathsf{false} \end{array}$$

Theorem (Loop detection by narrowing [FroCoS '05])

$(\mathcal{P}, \mathcal{R})$ is looping if there is some $\ell \to r \in \mathcal{P}$ such that

- $r \leadsto^*_{\mathcal{P},\mathcal{R},\sigma} \ell'$
- $\ell\sigma$ *matches* ℓ'

Example

$$
\begin{array}{rcl}
\mathsf{fact}^\sharp(x, y) & \to & \mathsf{if}^\sharp(x == y, x, y) \\
\mathsf{if}^\sharp(\mathsf{false}, x, y) & \to & \mathsf{fact}^\sharp(\mathsf{s}(x), y) \\
\hline
x == y & \to & \mathsf{chk}(\mathsf{eq}(x, y)) \\
\mathsf{eq}(x, x) & \to & \mathsf{true} \\
\mathsf{chk}(\mathsf{true}) & \to & \mathsf{true} \\
\mathsf{chk}(\mathsf{eq}(x, y)) & \to & \mathsf{false}
\end{array}
$$

- $r = \mathsf{if}^\sharp(x == y, x, y) \leadsto_{\mathcal{R},\{\}} \mathsf{if}^\sharp(\mathsf{chk}(\mathsf{eq}(x, y)), x, y)$
  $\leadsto_{\mathcal{R},\{\}} \mathsf{if}^\sharp(\mathsf{false}, x, y)$
  $\leadsto_{\mathcal{P},\{\}} \mathsf{fact}^\sharp(\mathsf{s}(x), y)$
- $\ell\sigma = \mathsf{fact}^\sharp(x, y)$ matches $\mathsf{fact}^\sharp(\mathsf{s}(x), y) = \ell'$

Theorem (Loop detection by narrowing [FroCoS '05])

$(\mathcal{P}, \mathcal{R})$ is looping if there is some $\ell \to r \in \mathcal{P}$ such that

- $r \rightsquigarrow^*_{\mathcal{P},\mathcal{R},\sigma} \ell'$
- $\ell\sigma$ *matches* ℓ'

Example

$$
\begin{array}{rcl}
\mathsf{fact}^\sharp(x,y) & \to & \mathsf{if}^\sharp(x == y, x, y) \\
\mathsf{if}^\sharp(\mathsf{false}, x, y) & \to & \mathsf{fact}^\sharp(\mathsf{s}(x), y) \\
\hline
x == y & \to & \mathsf{chk}(\mathsf{eq}(x,y)) \\
\mathsf{eq}(x,x) & \to & \mathsf{true} \\
\mathsf{chk}(\mathsf{true}) & \to & \mathsf{true} \\
\mathsf{chk}(\mathsf{eq}(x,y)) & \to & \mathsf{false}
\end{array}
$$

infinite $(\mathcal{P}, \mathcal{R})$-chain found,
termination disproven with counterexample

Outline

Disproving Termination

- Using Dependency Pairs
- Detecting Loops

Disproving Innermost Termination

- Decision Procedure to Detect Innermost Loops

Example

$$
\begin{array}{rcl}
\mathsf{fact}^\sharp(x,y) & \rightarrow & \mathsf{if}^\sharp(x == y, x, y) \\
\mathsf{if}^\sharp(\mathsf{false}, x, y) & \rightarrow & \mathsf{fact}^\sharp(\mathsf{s}(x), y) \\
\hline
x == y & \rightarrow & \mathsf{chk}(\mathsf{eq}(x,y)) \\
\mathsf{eq}(x,x) & \rightarrow & \mathsf{true} \\
\mathsf{chk}(\mathsf{true}) & \rightarrow & \mathsf{true} \\
\mathsf{chk}(\mathsf{eq}(x,y)) & \rightarrow & \mathsf{false}
\end{array}
$$

Example

$$\begin{array}{rcl} \mathsf{fact}^\sharp(x,y) & \to & \mathsf{if}^\sharp(x == y, x, y) \\ \mathsf{if}^\sharp(\mathsf{false}, x, y) & \to & \mathsf{fact}^\sharp(\mathsf{s}(x), y) \\ \hline x == y & \to & \mathsf{chk}(\mathsf{eq}(x,y)) \\ \mathsf{eq}(x,x) & \to & \mathsf{true} \\ \mathsf{chk}(\mathsf{true}) & \to & \mathsf{true} \\ \mathsf{chk}(\mathsf{eq}(x,y)) & \to & \mathsf{false} \end{array}$$

- Looping reduction

$$\begin{array}{rcl} \mathsf{fact}^\sharp(x,y) & \to_{\mathcal{P}} & \mathsf{if}^\sharp(x == y, x, y) \\ & \to_{\mathcal{R}} & \mathsf{if}^\sharp(\mathsf{chk}(\mathsf{eq}(x,y)), x, y) \\ & \to_{\mathcal{R}} & \mathsf{if}^\sharp(\mathsf{false}, x, y) \\ & \to_{\mathcal{P}} & \mathsf{fact}^\sharp(x,y)\{x/\mathsf{s}(x)\} \end{array}$$

Example

$$
\begin{array}{rcl}
\mathsf{fact}^\sharp(x,y) & \rightarrow & \mathsf{if}^\sharp(x == y, x, y) \\
\mathsf{if}^\sharp(\mathsf{false}, x, y) & \rightarrow & \mathsf{fact}^\sharp(\mathsf{s}(x), y) \\
\hline
x == y & \rightarrow & \mathsf{chk}(\mathsf{eq}(x,y)) \\
\mathsf{eq}(x,x) & \rightarrow & \mathsf{true} \\
\mathsf{chk}(\mathsf{true}) & \rightarrow & \mathsf{true} \\
\mathsf{chk}(\mathsf{eq}(x,y)) & \rightarrow & \mathsf{false}
\end{array}
$$

- Innermost looping?

$$
\begin{array}{rcl}
\mathsf{fact}^\sharp(x,y) & \rightarrow_{\mathcal{P}} & \mathsf{if}^\sharp(x == y, x, y) \\
 & \rightarrow_{\mathcal{R}} & \mathsf{if}^\sharp(\mathsf{chk}(\mathsf{eq}(x,y)), x, y) \\
 & \rightarrow_{\mathcal{R}} & \mathsf{if}^\sharp(\mathsf{false}, x, y) \\
 & \rightarrow_{\mathcal{P}} & \mathsf{fact}^\sharp(x,y)\{x/\mathsf{s}(x)\}
\end{array}
$$

Definition (Looping DP Problem)

$(\mathcal{P}, \mathcal{R})$ is looping iff there are t and μ such that
$t \to_{\mathcal{P}} \circ \to^*_{\mathcal{P},\mathcal{R}} t\mu$.
Infinite chain:

$$t \to_{\mathcal{P}} \circ \to^*_{\mathcal{P},\mathcal{R}} t\mu \to_{\mathcal{P}} \circ \to^*_{\mathcal{P},\mathcal{R}} t\mu^2 \to_{\mathcal{P}} \circ \to^*_{\mathcal{P},\mathcal{R}} t\mu^3 \ldots$$

Definition (Innermost Looping DP Problem?)

$(\mathcal{P}, \mathcal{R})$ is innermost looping iff there are t and μ such that
$t \xrightarrow{\mathrm{i}}_{\mathcal{P}} \circ \xrightarrow{\mathrm{i}}^{*}_{\mathcal{P},\mathcal{R}} t\mu$.
Infinite chain:

$$t \rightarrow_{\mathcal{P}} \circ \rightarrow^{*}_{\mathcal{P},\mathcal{R}} t\mu \rightarrow_{\mathcal{P}} \circ \rightarrow^{*}_{\mathcal{P},\mathcal{R}} t\mu^2 \rightarrow_{\mathcal{P}} \circ \rightarrow^{*}_{\mathcal{P},\mathcal{R}} t\mu^3 \ldots$$

Definition (Innermost Looping DP Problem?)

$(\mathcal{P}, \mathcal{R})$ is innermost looping iff there are t and μ such that $t \xrightarrow{\mathsf{i}}_{\mathcal{P}} \circ \xrightarrow{\mathsf{i}}^*_{\mathcal{P},\mathcal{R}} t\mu$.

Infinite innermost chain?:

$$t \xrightarrow{\mathsf{i}}_{\mathcal{P}} \circ \xrightarrow{\mathsf{i}}^*_{\mathcal{P},\mathcal{R}} t\mu \xrightarrow{\mathsf{i}}_{\mathcal{P}} \circ \xrightarrow{\mathsf{i}}^*_{\mathcal{P},\mathcal{R}} t\mu^2 \xrightarrow{\mathsf{i}}_{\mathcal{P}} \circ \xrightarrow{\mathsf{i}}^*_{\mathcal{P},\mathcal{R}} t\mu^3 \dots$$

Definition (Innermost Looping DP Problem?)

$(\mathcal{P}, \mathcal{R})$ is innermost looping iff there are t and μ such that
$t \xrightarrow{i}_{\mathcal{P}} \circ \xrightarrow{i}^{*}_{\mathcal{P},\mathcal{R}} t\mu$.
Infinite innermost chain?: No!

$$t \xrightarrow{i}_{\mathcal{P}} \circ \xrightarrow{i}^{*}_{\mathcal{P},\mathcal{R}} t\mu \xrightarrow{i}_{\mathcal{P}} \circ \xrightarrow{i}^{*}_{\mathcal{P},\mathcal{R}} t\mu^2 \xrightarrow{i}_{\mathcal{P}} \circ \xrightarrow{i}^{*}_{\mathcal{P},\mathcal{R}} t\mu^3 \ldots$$

Example

$$\mathsf{f}^\sharp(\mathsf{g}(x)) \rightarrow \mathsf{f}^\sharp(\mathsf{g}(\mathsf{g}(x)))$$

$$\mathsf{g}(\mathsf{g}(\mathsf{g}(x))) \rightarrow \mathsf{a}$$

"Innermost loop": (There is no infinite innermost $(\mathcal{P}, \mathcal{R})$-chain)

$$\begin{array}{rcl} t = \mathsf{f}^\sharp(\mathsf{g}(x)) & \xrightarrow{i}_{\mathcal{P}} & \mathsf{f}^\sharp(\mathsf{g}(\mathsf{g}(x))) = t\{x/\mathsf{g}(x)\} \\ & \xrightarrow{i}_{\mathcal{P}} & \mathsf{f}^\sharp(\mathsf{g}(\mathsf{g}(\mathsf{g}(x)))) \\ & \not\xrightarrow{i}_{\mathcal{P}} & \mathsf{f}^\sharp(\mathsf{g}(\mathsf{g}(\mathsf{g}(\mathsf{g}(x))))) \end{array}$$

Problem: $\xrightarrow{i}_{\mathcal{P},\mathcal{R}}$ is not closed under substitutions!

Definition (Looping DP Problem)

$(\mathcal{P}, \mathcal{R})$ is looping iff there are t and μ such that $t \to_{\mathcal{P}} \circ \to^*_{\mathcal{P},\mathcal{R}} t\mu$
iff there is a reduction

$$t = t_1 \to_{\ell_1 \to r_1, p_1} t_2 \to_{\ell_2 \to r_2, p_2} \ldots t_m \to_{\ell_m \to r_m, p_m} t\mu$$

Infinite chain

$$\begin{array}{llllllll}
t & = t_1 & \to_{\ell_1 \to r_1, p_1} & t_2 & \to_{\ell_2 \to r_2, p_2} & \ldots & t_m & \to_{\ell_m \to r_m, p_m} \\
t\mu & = t_1\mu & \to_{\ell_1 \to r_1, p_1} & t_2\mu & \to_{\ell_2 \to r_2, p_2} & \ldots & t_m\mu & \to_{\ell_m \to r_m, p_m} \\
t\mu^2 & = t_1\mu^2 & \to_{\ell_1 \to r_1, p_1} & t_2\mu^2 & \to_{\ell_2 \to r_2, p_2} & \ldots & t_m\mu^2 & \to_{\ell_m \to r_m, p_m} \\
t\mu^3 & = t_1\mu^3 & \to_{\ell_1 \to r_1, p_1} & t_2\mu^3 & \to_{\ell_2 \to r_2, p_2} & \ldots & &
\end{array}$$

Definition (Looping DP Problem)

$(\mathcal{P}, \mathcal{R})$ is looping iff there are t and μ such that $t \rightarrow_{\mathcal{P}} \circ \rightarrow^*_{\mathcal{P},\mathcal{R}} t\mu$
iff there is a reduction

$$t = t_1 \rightarrow_{\ell_1 \rightarrow r_1, p_1} t_2 \rightarrow_{\ell_2 \rightarrow r_2, p_2} \ldots t_m \rightarrow_{\ell_m \rightarrow r_m, p_m} t\mu$$

Infinite innermost chain
iff all direct subterms of all $t_i \mu^n|_{p_i}$ are in $\mathcal{R}$-normal form

$$\begin{array}{llllllll}
t & = t_1 & \rightarrow_{\ell_1 \rightarrow r_1, p_1} & t_2 & \rightarrow_{\ell_2 \rightarrow r_2, p_2} & \ldots & t_m & \rightarrow_{\ell_m \rightarrow r_m, p_m} \\
t\mu & = t_1\mu & \rightarrow_{\ell_1 \rightarrow r_1, p_1} & t_2\mu & \rightarrow_{\ell_2 \rightarrow r_2, p_2} & \ldots & t_m\mu & \rightarrow_{\ell_m \rightarrow r_m, p_m} \\
t\mu^2 & = t_1\mu^2 & \rightarrow_{\ell_1 \rightarrow r_1, p_1} & t_2\mu^2 & \rightarrow_{\ell_2 \rightarrow r_2, p_2} & \ldots & t_m\mu^2 & \rightarrow_{\ell_m \rightarrow r_m, p_m} \\
t\mu^3 & = t_1\mu^3 & \rightarrow_{\ell_1 \rightarrow r_1, p_1} & t_2\mu^3 & \rightarrow_{\ell_2 \rightarrow r_2, p_2} & \ldots & &
\end{array}$$

Definition (Innermost Looping DP Problem)

$(\mathcal{P}, \mathcal{R})$ is innermost looping iff
iff there is a reduction

$$t = t_1 \to_{\ell_1 \to r_1, p_1} t_2 \to_{\ell_2 \to r_2, p_2} \ldots t_m \to_{\ell_m \to r_m, p_m} t\mu$$

such that all direct subterms of all $t_i \mu^n|_{p_i}$ are in $\mathcal{R}$-normal form

Infinite innermost chain

$$\begin{array}{llllllll}
t & = t_1 & \xrightarrow{\text{i}}_{\ell_1 \to r_1, p_1} & t_2 & \xrightarrow{\text{i}}_{\ell_2 \to r_2, p_2} & \ldots & t_m & \xrightarrow{\text{i}}_{\ell_m \to r_m, p_m} \\
t\mu & = t_1\mu & \xrightarrow{\text{i}}_{\ell_1 \to r_1, p_1} & t_2\mu & \xrightarrow{\text{i}}_{\ell_2 \to r_2, p_2} & \ldots & t_m\mu & \xrightarrow{\text{i}}_{\ell_m \to r_m, p_m} \\
t\mu^2 & = t_1\mu^2 & \xrightarrow{\text{i}}_{\ell_1 \to r_1, p_1} & t_2\mu^2 & \xrightarrow{\text{i}}_{\ell_2 \to r_2, p_2} & \ldots & t_m\mu^2 & \xrightarrow{\text{i}}_{\ell_m \to r_m, p_m} \\
t\mu^3 & = t_1\mu^3 & \xrightarrow{\text{i}}_{\ell_1 \to r_1, p_1} & t_2\mu^3 & \xrightarrow{\text{i}}_{\ell_2 \to r_2, p_2} & \ldots & &
\end{array}$$

Definition (Innermost Looping DP Problem / TRS)

$(\mathcal{P}, \mathcal{R})$ / $\mathcal{R}$ is innermost looping iff
iff there is a reduction

$$t = t_1 \to_{\ell_1 \to r_1, p_1} t_2 \to_{\ell_2 \to r_2, p_2} \ldots t_m \to_{\ell_m \to r_m, p_m} t\mu \;/\; C[t\mu]$$

such that all direct subterms of all $t_i\mu^n|_{p_i}$ are in $\mathcal{R}$-normal form

Corresponding definition also possible for innermost looping TRSs

New Theorem (Equivalence of Innermost Loopingness)

$\mathcal{R}$ innermost looping iff $(DP(\mathcal{R}), \mathcal{R})$ innermost looping

Definition (Innermost Looping DP Problem / TRS)

$(\mathcal{P},\mathcal{R})$ / $\mathcal{R}$ is innermost looping iff
iff there is a reduction

$$t = t_1 \to_{\ell_1\to r_1,p_1} t_2 \to_{\ell_2\to r_2,p_2} \dots t_m \to_{\ell_m\to r_m,p_m} t\mu \;/\; C[t\mu]$$

such that all direct subterms of all $t_i\mu^n|_{p_i}$ are in $\mathcal{R}$-normal form

Problem

How to check "all direct subterms of all $t_i\mu^n|_{p_i}$ are in $\mathcal{R}$-normal form"?

- $t_i|_{p_i} = f(s_1,\dots,s_k)$. Thus, "all direct subterms of $t_i\mu^n|_{p_i}$" are all $s_j\mu^n$
- $s\mu^n$ in $\mathcal{R}$-normal form iff no $\ell\to r\in\mathcal{R}$ such that $s\mu^n|_p = \ell\sigma$

Definition (Innermost Looping DP Problem / TRS)

$(\mathcal{P}, \mathcal{R})$ / $\mathcal{R}$ is innermost looping iff
iff there is a reduction

$$t = t_1 \to_{\ell_1 \to r_1, p_1} t_2 \to_{\ell_2 \to r_2, p_2} \ldots t_m \to_{\ell_m \to r_m, p_m} t\mu \ / \ C[t\mu]$$

such that all direct subterms of all $t_i \mu^n|_{p_i}$ are in $\mathcal{R}$-normal form

Problem

How to check "all direct subterms of all $t_i \mu^n|_{p_i}$ are in $\mathcal{R}$-normal form"?

- $t_i|_{p_i} = f(s_1, \ldots, s_k)$. Thus, "all direct subterms of $t_i \mu^n|_{p_i}$" are all $s_j \mu^n$
- $s\mu^n$ in $\mathcal{R}$-normal form iff no $\ell \to r \in \mathcal{R}$ such that $s\mu^n|_p = \ell\sigma$

$\Rightarrow$ Innermost loopingness reducible to redex problems:

Given $(s \mid\!\!> \ell, \mu)$, are there n, σ, p such that $s\mu^n|_p = \ell\sigma$?

Definition (Innermost Looping DP Problem / TRS)

$(\mathcal{P}, \mathcal{R})$ / $\mathcal{R}$ is innermost looping iff
iff there is a reduction

$$t = t_1 \to_{\ell_1 \to r_1, p_1} t_2 \to_{\ell_2 \to r_2, p_2} \ldots t_m \to_{\ell_m \to r_m, p_m} t\mu \;/\; C[t\mu]$$

such that all direct subterms of all $t_i\mu^n|_{p_i}$ are in $\mathcal{R}$-normal form

Problem

How to check "all direct subterms of all $t_i\mu^n|_{p_i}$ are in $\mathcal{R}$-normal form"?

- $t_i|_{p_i} = f(s_1, \ldots, s_k)$. Thus, "all direct subterms of $t_i\mu^n|_{p_i}$" are all $s_j\mu^n$
- $s\mu^n$ in $\mathcal{R}$-normal form iff no $\ell \to r \in \mathcal{R}$ such that $s\mu^n|_p = \ell\sigma$ iff redex problem $(s \mid\!\gtrdot \ell, \mu)$ not solvable

Example

$$\begin{array}{rcl} t_1 = \mathsf{f}^\sharp(\mathsf{g}(x)) & \to & \mathsf{f}^\sharp(\mathsf{g}(\mathsf{g}(x))) = t_1\{x/\mathsf{g}(x)\} \\ \mathsf{g}(\mathsf{g}(\mathsf{g}(x))) & \to & \mathsf{a} \end{array}$$

Innermost loop iff redex problem $(\mathsf{g}(x) \mid\!\gtrdot \mathsf{g}(\mathsf{g}(\mathsf{g}(x))), \{x/\mathsf{g}(x)\})$ not s.

Decision Procedure to Detect Innermost Loops

Definition (Innermost Looping DP Problem / TRS)

$(\mathcal{P}, \mathcal{R})$ / $\mathcal{R}$ is innermost looping iff there is a reduction

$$t = t_1 \to_{\ell_1 \to r_1, p_1} t_2 \to_{\ell_2 \to r_2, p_2} \ldots t_m \to_{\ell_m \to r_m, p_m} t\mu \;/\; C[t\mu]$$

such that all direct subterms of all $t_i \mu^n|_{p_i}$ are in $\mathcal{R}$-normal form.

transform into set of

Definition (Redex Problem)

A redex problem is a triple $(s \mid\!\!\geqslant \ell, \mu)$. Solvable iff

there are n, σ, p such that $s\mu^n|_p = \ell\sigma$

Definition (Redex Problem)

A redex problem is a triple $(s \mathrel{|\!\!\gtrdot} \ell, \mu)$. Solvable iff

there are n, σ, p such that $s\mu^n|_p = \ell\sigma$

transform into set of

Definition (Matching Problem)

A matching problem is a triple $(s \gtrdot \ell, \mu)$. Solvable iff

there are n, σ such that $s\mu^n = \ell\sigma$

transform into set of

Definition (Identity Problem)

An identity problem is a triple $(s \approxeq t, \mu)$. Solvable iff

there is an n such that $s\mu^n = t\mu^n$

and finally decide solvability of identity problems

Example (Redex Problems, $(s \mid\!\!> \ell, \mu)$ solvable iff $s\mu^n|_p = \ell\sigma$)

$$\begin{aligned}
\mathsf{fact}^\sharp(x, y) &\to \mathsf{if}^\sharp(x == y, x, y) \\
\mathsf{if}^\sharp(\mathsf{false}, x, y) &\to \mathsf{fact}^\sharp(\mathsf{s}(x), y)
\end{aligned}$$

$$\begin{aligned}
x == y &\to \mathsf{chk}(\mathsf{eq}(x, y)) \\
\mathsf{eq}(x, x) &\to \mathsf{true} \\
\mathsf{chk}(\mathsf{true}) &\to \mathsf{true} \\
\mathsf{chk}(\mathsf{eq}(x, y)) &\to \mathsf{false}
\end{aligned}$$

Checking innermost loop with reduction

$$\mathsf{if}^\sharp(\mathsf{chk}(\mathsf{eq}(x, y)), x, y) \to_{\mathcal{R}} \mathsf{if}^\sharp(\mathsf{false}, x, y)$$

results in redex problem $(\mathsf{eq}(x, y) \mid\!\!> \mathsf{eq}(x, x), \mu)$ with $\mu = \{x/\mathsf{s}(x)\}$

Example (Redex Problems, $(s \mid\!\gtrdot \ell, \mu)$ solvable iff $s\mu^n|_p = \ell\sigma$)

Handle $(\mathsf{eq}(x, y) \mid\!\gtrdot \mathsf{eq}(x, x), \mu)$ with $\mu = \{x/\mathsf{s}(x)\}$

New Theorem (Transforming Redex Problems into Matching Problems)

Let $\mathcal{W} = \bigcup_{n \in \mathbb{N}} \mathcal{V}(s\mu^n)$. Then the following statements are equivalent.

- $(s \mid\!\gtrdot \ell, \mu)$ is solvable
- for a non-variable subterm u of a term in $\{s\} \cup \{x\mu \mid x \in \mathcal{W}\}$ the matching problem $(u \gtrdot \ell, \mu)$ is solvable

Example (Redex Problems, $(s \mid\!\gtrdot \ell, \mu)$ solvable iff $s\mu^n|_p = \ell\sigma$)

Handle $(\mathsf{eq}(x, y) \mid\!\gtrdot \mathsf{eq}(x, x), \mu)$ with $\mu = \{x/\mathsf{s}(x)\}$

n	$s\mu^n$	$\mathcal{V}(s\mu^n)$
0	$\mathsf{eq}(x, y)$	$\{x, y\}$
1	$\mathsf{eq}(\mathsf{s}(x), y)$	$\{x, y\}$

Hence, $\mathcal{W} = \{x, y\}$ and u is subterm of term in $\{\mathsf{eq}(x, y), \mathsf{s}(x), y\}$.

New Theorem (Transforming Redex Problems into Matching Problems)

Let $\mathcal{W} = \bigcup_{n \in \mathbb{N}} \mathcal{V}(s\mu^n)$. Then the following statements are equivalent.

- $(s \mid\!\gtrdot \ell, \mu)$ is solvable
- for a non-variable subterm u of a term in $\{s\} \cup \{x\mu \mid x \in \mathcal{W}\}$ the matching problem $(u \gtrdot \ell, \mu)$ is solvable

Example (Redex Problems, $(s \mathrel{|\!\gtrdot} \ell, \mu)$ solvable iff $s\mu^n|_p = \ell\sigma$)

Handle $(\mathsf{eq}(x, y) \mathrel{|\!\gtrdot} \mathsf{eq}(x, x), \mu)$ with $\mu = \{x/\mathsf{s}(x)\}$

Matching problems: $(u \gtrdot \ell, \mu)$ for all non-variable subterms u

$$(\mathsf{eq}(x, y) \gtrdot \mathsf{eq}(x, x), \mu) \quad \text{and} \quad (\mathsf{s}(x) \gtrdot \mathsf{eq}(x, x), \mu)$$

Hence, $\mathcal{W} = \{x, y\}$ and u is subterm of term in $\{\mathsf{eq}(x, y), \mathsf{s}(x), y\}$.

New Theorem (Transforming Redex Problems into Matching Problems)

Let $\mathcal{W} = \bigcup_{n \in \mathbb{N}} \mathcal{V}(s\mu^n)$. Then the following statements are equivalent.

- $(s \mathrel{|\!\gtrdot} \ell, \mu)$ is solvable
- for a non-variable subterm u of a term in $\{s\} \cup \{x\mu \mid x \in \mathcal{W}\}$ the matching problem $(u \gtrdot \ell, \mu)$ is solvable

Definition (Matching Problem)

A matching problem $(s \gtrdot \ell, \mu)$ is solvable iff $s\mu^n = \ell\sigma$.

transform into set of

Definition (Matching Problem)

An identity problem $(s \approxeq t, \mu)$ is solvable iff $s\mu^n = t\mu^n$.

Example

- Substitution $\mu = \{x/\mathsf{s}(x)\}$
- $(\mathsf{eq}(x, y) \gtrdot \mathsf{eq}(x, x), \mu)$ solvable iff $(x \approxeq y, \mu)$ solvable
- $(\mathsf{s}(x) \gtrdot \mathsf{eq}(x, x), \mu)$ not solvable

Algorithm for Identity Problems ($(s \approx t, \mu)$ solvable iff $\exists n : s\mu^n = t\mu^n$)

2. If $s = t$ then stop with result "Yes"

8. $s := s\mu, t := t\mu$
9. Continue with step 2

All solvable identity problems will be detected!

Algorithm for Identity Problems ($(s \approx t, \mu)$ solvable iff $\exists n : s\mu^n = t\mu^n$)

2. If $s = t$ then stop with result "Yes"
3. If there is a shared position p of s and t such that $s|_p = f(\ldots)$ and $t|_p = g(\ldots)$ and $f \neq g$ then stop with result "No"

8. $s := s\mu, t := t\mu$
9. Continue with step 2

$$s = \mathsf{h}(\mathsf{f}(y), \mathsf{f}(z)), t = \mathsf{h}(\mathsf{g}(x), \mathsf{f}(\mathsf{f}(\mathsf{a}))), \mu = \{\ldots\}$$

Algorithm for Identity Problems ($(s \approx t, \mu)$ solvable iff $\exists n : s\mu^n = t\mu^n$)

2. If $s = t$ then stop with result "Yes"
3. If there is a shared position p of s and t such that $s|_p = f(\ldots)$ and $t|_p = g(\ldots)$ and $f \neq g$ then stop with result "No"
4. If there is a shared position p of s and t such that $s|_p = f(\ldots)$, $t|_p = x \notin Dom(\mu)$ then stop with result "No" (sym.)

8. $s := s\mu, t := t\mu$
9. Continue with step 2

$$s = \mathsf{h}(\mathsf{f}(y), \mathsf{f}(z)), t = \mathsf{h}(x, \mathsf{f}(\mathsf{f}(\mathsf{a}))), \mu = \{y/\mathsf{f}(y), z/\mathsf{f}(\mathsf{a})\}$$

Algorithm for Identity Problems ($(s \approx t, \mu)$ solvable iff $\exists n : s\mu^n = t\mu^n$)

2. If $s = t$ then stop with result "Yes"
3. If there is a shared position p of s and t such that $s|_p = f(\dots)$ and $t|_p = g(\dots)$ and $f \neq g$ then stop with result "No"
4. If there is a shared position p of s and t such that $s|_p = f(\dots)$, $t|_p = x \notin Dom(\mu)$ then stop with result "No" (sym.)
5. If there is a shared position p of s and t such that $s|_p = x$, $t|_p = y$, $x, y \notin Dom(\mu)$, and $x \neq y$ then stop with result "No"
8. $s := s\mu, t := t\mu$
9. Continue with step 2

$$s = \mathsf{h}(x, \mathsf{f}(z)), t = \mathsf{h}(y, \mathsf{f}(\mathsf{f}(\mathsf{a}))), \mu = \{z/\mathsf{f}(\mathsf{a})\}$$

Algorithm for Identity Problems ($(s \approx t, \mu)$ solvable iff $\exists n : s\mu^n = t\mu^n$)

2. If $s = t$ then stop with result "Yes"
3. If there is a shared position p of s and t such that $s|_p = f(\dots)$ and $t|_p = g(\dots)$ and $f \neq g$ then stop with result "No"
4. If there is a shared position p of s and t such that $s|_p = f(\dots)$, $t|_p = x \notin Dom(\mu)$ then stop with result "No" (sym.)
5. If there is a shared position p of s and t such that $s|_p = x$, $t|_p = y$, $x, y \notin Dom(\mu)$, and $x \neq y$ then stop with result "No"
8. $s := s\mu, t := t\mu$
9. Continue with step 2

Example (Identity problem $(x \approx y, \{x/\mathsf{s}(x)\})$)

In second iteration: $s = \mathsf{s}(x)$ and $t = y$.
Result is "No" due to 4

Algorithm for Identity Problems ($(s \approx t, \mu)$ solvable iff $\exists n : s\mu^n = t\mu^n$)

2. If $s = t$ then stop with result "Yes"
3. If there is a shared position p of s and t such that $s|_p = f(\dots)$ and $t|_p = g(\dots)$ and $f \neq g$ then stop with result "No"
4. If there is a shared position p of s and t such that $s|_p = f(\dots)$, $t|_p = x \notin Dom(\mu)$ then stop with result "No" (sym.)
5. If there is a shared position p of s and t such that $s|_p = x$, $t|_p = y$, $x, y \notin Dom(\mu)$, and $x \neq y$ then stop with result "No"
8. $s := s\mu, t := t\mu$
9. Continue with step 2

Up to now:
detect identity problems which are solvable or which satisfy $s\mu^\infty \neq t\mu^\infty$

Consider $s = x, t = y, \mu = \{x/\mathsf{f}(y), y/\mathsf{f}(x)\}$

Identity problems is not solvable, but $s\mu^\infty = \mathsf{f}(\mathsf{f}(\dots)) = t\mu^\infty$
$\Rightarrow$ Add missing steps to detect all non-solvable problems

Algorithm for Identity Problems ($(s \approx t, \mu)$ solvable iff $\exists n : s\mu^n = t\mu^n$)

1. $\mathcal{C} := \emptyset$; While $\mu = \{x_1/x_2, \ldots, x_n/x_1, \ldots\}, n > 1$ do $\mu := \mu^n$
2. If $s = t$ then stop with result "Yes"
3. If there is a shared position p of s and t such that $s|_p = f(\ldots)$ and $t|_p = g(\ldots)$ and $f \neq g$ then stop with result "No"
4. If there is a shared position p of s and t such that $s|_p = f(\ldots)$, $t|_p = x \notin Dom(\mu)$ then stop with result "No" (sym.)
5. If there is a shared position p of s and t such that $s|_p = x$, $t|_p = y$, $x, y \notin Dom(\mu)$, and $x \neq y$ then stop with result "No"
6. Add the triple $(x, p, t|_p)$ to $\mathcal{C}$ for all shared positions p of s and t such that $x = s|_p \neq t|_p$ where x is an increasing variable (sym.)
7. If $(x, p_1, u_1) \in \mathcal{C}$ and $(x, p_2, u_2) \in \mathcal{C}$ where
   - u_1 and u_2 are not unifiable or where
   - $u_1 = u_2$ and $p_1 < p_2$

   then stop with result "No"
8. $s := s\mu, t := t\mu$
9. Continue with step 2

New Theorem (Deciding solvability of identity problems)

The algorithm for identity problems terminates and is correct.

New Theorem (Deciding solvability of identity problems)

The algorithm for identity problems terminates and is correct.

Example ($\mu = \{x/\mathsf{s}(x)\}$)

- $\Rightarrow$ $(x \approxeq y, \mu)$ not solvable
- $\Rightarrow$ $(\mathsf{eq}(x, y) \gtrdot \mathsf{eq}(x, x), \mu)$ not solvable
- $\Rightarrow$ $(\mathsf{eq}(x, y) \mid\!\gtrdot \mathsf{eq}(x, x), \mu)$ not solvable
- $\Rightarrow$ looping reduction is innermost looping
- $\Rightarrow$ innermost termination disproven with counterexample

New Theorem (Deciding solvability of identity problems)

The algorithm for identity problems terminates and is correct.

Example ($\mu = \{x/\mathsf{s}(x)\}$)

- $\Rightarrow$ $(x \approxeq y, \mu)$ not solvable
- $\Rightarrow$ $(\mathsf{eq}(x, y) \gtrdot \mathsf{eq}(x, x), \mu)$ not solvable
- $\Rightarrow$ $(\mathsf{eq}(x, y) \mid\gtrdot \mathsf{eq}(x, x), \mu)$ not solvable
- $\Rightarrow$ looping reduction is innermost looping
- $\Rightarrow$ innermost termination disproven with counterexample

New Theorem (Deciding solvability of identity problems)

The algorithm for identity problems terminates and is correct.

Example ($\mu = \{x/\mathsf{s}(x)\}$)

- $\Rightarrow$ $(x \approxeq y, \mu)$ not solvable
- $\Rightarrow$ $(\mathsf{eq}(x, y) \gtrdot \mathsf{eq}(x, x), \mu)$ not solvable
- $\Rightarrow$ $(\mathsf{eq}(x, y) \mid\!\gtrdot \mathsf{eq}(x, x), \mu)$ not solvable
- $\Rightarrow$ looping reduction is innermost looping
- $\Rightarrow$ innermost termination disproven with counterexample

New Theorem (Deciding solvability of identity problems)

The algorithm for identity problems terminates and is correct.

Example ($\mu = \{x/\mathsf{s}(x)\}$)

- $\Rightarrow$ $(x \approxeq y, \mu)$ not solvable
- $\Rightarrow$ $(\mathsf{eq}(x, y) \gtrdot \mathsf{eq}(x, x), \mu)$ not solvable
- $\Rightarrow$ $(\mathsf{eq}(x, y) \mid\!\gtrdot \mathsf{eq}(x, x), \mu)$ not solvable
- $\Rightarrow$ looping reduction is innermost looping
- $\Rightarrow$ innermost termination disproven with counterexample

New Theorem (Deciding solvability of identity problems)

The algorithm for identity problems terminates and is correct.

Example ($\mu = \{x/\mathsf{s}(x)\}$)

- $\Rightarrow$ $(x \approxeq y, \mu)$ not solvable
- $\Rightarrow$ $(\mathsf{eq}(x,y) \gtrdot \mathsf{eq}(x,x), \mu)$ not solvable
- $\Rightarrow$ $(\mathsf{eq}(x,y) \mid\gtrdot \mathsf{eq}(x,x), \mu)$ not solvable
- $\Rightarrow$ looping reduction is innermost looping
- $\Rightarrow$ innermost termination disproven with counterexample

Experimental data

- Implemented in termination prover AProVE
- Results at annual international Termination Competition
  975 TRSs of Termination Problem DataBase
  (≤ 242 non-terminating, ≤ 179 innermost non-terminating)

	AProVE	NTI	TTT2	Jambox	Σ
non-term.	128	117	69	16	129
innermost non-term.	95	not supported			95

Experimental data

- Implemented in termination prover AProVE
- Results at annual international Termination Competition
  975 TRSs of Termination Problem DataBase
  (≤ 242 non-terminating, ≤ 179 innermost non-terminating)

	AProVE	NTI	TTT2	Jambox	Σ
non-term.	128	117	69	16	129
innermost non-term.	95	not supported			95

All but 5 of the $128 - 95$ TRSs are
not terminating, but innermost terminating

Summary

- Dependency pairs techniques identify non-terminating parts
- Detect loops by narrowing
- New decision procedure to detect innermost loops
  (works on TRSs and DP problems)
- Implemented in termination prover AProVE
  `http://aprove.informatik.rwth-aachen.de`
- Outlook
  - Complexity of solving identity problems
  - Prune search tree for loop detection
    Conditions under which narrowing can find all loops
  - Disprove (innermost) termination of non-looping TRSs

Summary

- Dependency pairs techniques identify non-terminating parts
- Detect loops by narrowing
- New decision procedure to detect innermost loops
  (works on TRSs and DP problems)

- Implemented in termination prover AProVE
  `http://aprove.informatik.rwth-aachen.de`

- Outlook
  - Complexity of solving identity problems
  - Prune search tree for loop detection
    Conditions under which narrowing can find all loops
  - Disprove (innermost) termination of non-looping TRSs

Summary

- Dependency pairs techniques identify non-terminating parts
- Detect loops by narrowing
- New decision procedure to detect innermost loops
  (works on TRSs and DP problems)
- Implemented in termination prover AProVE
  `http://aprove.informatik.rwth-aachen.de`
- Outlook
  - Complexity of solving identity problems
  - Prune search tree for loop detection
    Conditions under which narrowing can find all loops
  - Disprove (innermost) termination of non-looping TRSs

Example (Matching Problems, $(\{s_1 \gtrdot \ell_1 \ldots s_m \gtrdot \ell_m\}, \mu)$ solvable iff $s_i\mu^n = \ell_i\sigma$ for all i)

- $\mu = \{x/\mathsf{s}(x)\}$

$$\begin{array}{r}(\{\mathsf{eq}(x,y) \gtrdot \mathsf{eq}(x,x)\}, \mu) \\ (\{\mathsf{s}(x) \gtrdot \mathsf{eq}(x,x)\}, \mu)\end{array}$$

Example (Matching Problems, $(\{s_1 \gtrdot \ell_1 \dots s_m \gtrdot \ell_m\}, \mu)$ solvable iff $s_i\mu^n = \ell_i\sigma$ for all i)

- $\mu = \{x/\mathsf{s}(x)\}$

$$(\{\mathsf{eq}(x,y) \gtrdot \mathsf{eq}(x,x)\}, \mu)$$
$$(\{\mathsf{s}(x) \gtrdot \mathsf{eq}(x,x)\}, \mu)$$

x is increasing variable ($x \in \mathcal{V}_{incr}$) iff $x\mu^\infty \notin \mathcal{V}$ (here, $\mathcal{V}_{incr} = \{x\}$)

Definition (Transforming of Matching Problems)

$(\mathcal{M}, \mu)$ is transformed if $\mathcal{M} = \mathcal{M}' \uplus \{s \gtrdot f(\ell_1, \dots, \ell_k)\}$

Example (Matching Problems, $(\{s_1 \gtrdot \ell_1 \dots s_m \gtrdot \ell_m\}, \mu)$ solvable iff $s_i \mu^n = \ell_i \sigma$ for all i)

- $\mu = \{x / \mathsf{s}(x)\}$

$$(\{\mathsf{eq}(x, y) \gtrdot \mathsf{eq}(x, x)\}, \mu) \Rightarrow (\{x \gtrdot x, y \gtrdot x\}, \mu)$$
$$(\{\mathsf{s}(x) \gtrdot \mathsf{eq}(x, x)\}, \mu)$$

x is increasing variable ($x \in \mathcal{V}_{incr}$) iff $x\mu^\infty \notin \mathcal{V}$ (here, $\mathcal{V}_{incr} = \{x\}$)

Definition (Transforming of Matching Problems)

$(\mathcal{M}, \mu)$ is transformed if $\mathcal{M} = \mathcal{M}' \uplus \{s \gtrdot f(\ell_1, \dots, \ell_k)\}$

1. $\mathcal{M} \Rightarrow \mathcal{M}' \cup \{s_1 \gtrdot \ell_1, \dots, s_k \gtrdot \ell_k\}$, if $s = f(s_1, \dots, s_k)$

Example (Matching Problems, $(\{s_1 \gtrdot \ell_1 \dots s_m \gtrdot \ell_m\}, \mu)$ solvable iff $s_i \mu^n = \ell_i \sigma$ for all i)

- $\mu = \{x/\mathsf{s}(x)\}$

$$(\{\mathsf{eq}(x,y) \gtrdot \mathsf{eq}(x,x)\}, \mu) \Rightarrow (\{x \gtrdot x, y \gtrdot x\}, \mu)$$
$$(\{\mathsf{s}(x) \gtrdot \mathsf{eq}(x,x)\}, \mu) \Rightarrow \quad \bot$$

x is increasing variable ($x \in \mathcal{V}_{incr}$) iff $x\mu^\infty \notin \mathcal{V}$ (here, $\mathcal{V}_{incr} = \{x\}$)

Definition (Transforming of Matching Problems)

$(\mathcal{M}, \mu)$ is transformed if $\mathcal{M} = \mathcal{M}' \uplus \{s \gtrdot f(\ell_1, \dots, \ell_k)\}$

1. $\mathcal{M} \Rightarrow \mathcal{M}' \cup \{s_1 \gtrdot \ell_1, \dots, s_k \gtrdot \ell_k\}$, if $s = f(s_1, \dots, s_k)$
2. $\mathcal{M} \Rightarrow \bot$, if $s = g(\dots)$ and $f \neq g$

Example (Matching Problems, $(\{s_1 \gtrdot \ell_1 \dots s_m \gtrdot \ell_m\}, \mu)$ solvable iff $s_i\mu^n = \ell_i\sigma$ for all i)

- $\mu = \{x/\mathsf{s}(x)\}$

$$(\{\mathsf{eq}(x,y) \gtrdot \mathsf{eq}(x,x)\}, \mu) \Rightarrow (\{x \gtrdot x, y \gtrdot x\}, \mu)$$
$$(\{\mathsf{s}(x) \gtrdot \mathsf{eq}(x,x)\}, \mu) \Rightarrow \bot$$

x is increasing variable ($x \in \mathcal{V}_{incr}$) iff $x\mu^\infty \notin \mathcal{V}$ (here, $\mathcal{V}_{incr} = \{x\}$)

Definition (Transforming of Matching Problems)

$(\mathcal{M}, \mu)$ is transformed if $\mathcal{M} = \mathcal{M}' \uplus \{s \gtrdot f(\ell_1, \dots, \ell_k)\}$

1. $\mathcal{M} \Rightarrow \mathcal{M}' \cup \{s_1 \gtrdot \ell_1, \dots, s_k \gtrdot \ell_k\}$, if $s = f(s_1, \dots, s_k)$
2. $\mathcal{M} \Rightarrow \bot$, if $s = g(\dots)$ and $f \neq g$
3. $\mathcal{M} \Rightarrow \{s'\mu \gtrdot \ell' \mid s' \gtrdot \ell' \in \mathcal{M}\}$, if $s \in \mathcal{V}_{incr}$
4. $\mathcal{M} \Rightarrow \bot$, if $s \in \mathcal{V} \setminus \mathcal{V}_{incr}$

3. $x \gtrdot \mathsf{s}(y) \Rightarrow \mathsf{s}(x) \gtrdot \mathsf{s}(y)$
4. $y \gtrdot \mathsf{s}(x) \Rightarrow \bot$

Example (Matching Problems, $(\{s_1 \gtrdot \ell_1 \dots s_m \gtrdot \ell_m\}, \mu)$ solvable iff $s_i \mu^n = \ell_i \sigma$ for all i)

- $\mu = \{x / \mathsf{s}(x)\}$

$$
\begin{array}{rcl}
(\{\mathsf{eq}(x,y) \gtrdot \mathsf{eq}(x,x)\}, \mu) & \Rightarrow & (\{x \gtrdot x, y \gtrdot x\}, \mu) \\
(\{\mathsf{s}(x) \gtrdot \mathsf{eq}(x,x)\}, \mu) & \Rightarrow & \bot
\end{array}
$$

New Theorem (Transforming Matching- into Identity Problems)

- $\Rightarrow$ is confluent and terminating.
- If $(\mathcal{M}, \mu) \Downarrow = \bot$ then $(\mathcal{M}, \mu)$ is not solvable.
- If $(\mathcal{M}, \mu) \Downarrow = (\mathcal{M}', \mu)$ then
  - $\mathcal{M}' = \{s_1 \gtrdot x_1, \dots, s_k \gtrdot x_k\}$
  - $(\mathcal{M}, \mu)$ is solvable iff for all i, j with $x_i = x_j, s_i \neq s_j$ the identity problem $(s_i \approxeq s_j, \mu)$ is solvable.

Example (Matching Problems, $(\{s_1 \gtrdot \ell_1 \dots s_m \gtrdot \ell_m\}, \mu)$ solvable iff $s_i\mu^n = \ell_i\sigma$ for all i)

- $\mu = \{x/\mathsf{s}(x)\}$

$$(\{\mathsf{eq}(x,y) \gtrdot \mathsf{eq}(x,x)\}, \mu) \Rightarrow (\{x \gtrdot x, y \gtrdot x\}, \mu)$$
$$(\{\mathsf{s}(x) \gtrdot \mathsf{eq}(x,x)\}, \mu) \Rightarrow \bot$$

Resulting identity problem: $(x \approxeq y, \mu)$

New Theorem (Transforming Matching- into Identity Problems)

- $\Rightarrow$ is confluent and terminating.
- If $(\mathcal{M}, \mu) \Downarrow = \bot$ then $(\mathcal{M}, \mu)$ is not solvable.
- If $(\mathcal{M}, \mu) \Downarrow = (\mathcal{M}', \mu)$ then
  - $\mathcal{M}' = \{s_1 \gtrdot x_1, \dots, s_k \gtrdot x_k\}$
  - $(\mathcal{M}, \mu)$ is solvable iff for all i, j with $x_i = x_j, s_i \neq s_j$ the identity problem $(s_i \approxeq s_j, \mu)$ is solvable.

backward narrowing fails

$$\begin{array}{rcl} \mathsf{f}^\sharp(x,x) & \to & \mathsf{f}^\sharp(\;2\;,\;3\;) \\ 2 & \to & \mathsf{b} \\ 3 & \to & \mathsf{b} \end{array}$$

narrowing fails

$$\begin{array}{rcl} \mathsf{g}^\sharp(\;0\;,\;1\;,x) & \to & \mathsf{g}^\sharp(x,x,x) \\ \mathsf{a} & \to & 0 \\ \mathsf{a} & \to & 1 \end{array}$$

narrowing and backward narrowing fail

$$\begin{array}{rcl} \mathsf{f}^\sharp(x,x,y) & \to & \mathsf{f}^\sharp(\;2(y,y)\;,\;3(y,y)\;,y) \\ 2(\;0\;,\;1\;) & \to & \mathsf{b} \\ 3(\;0\;,\;1\;) & \to & \mathsf{b} \\ \mathsf{a} & \to & 0 \\ \mathsf{a} & \to & 1 \end{array}$$

overlap closures fail

$$\begin{array}{rcl} \mathsf{f}^\sharp(x,y,x,y,z) & \to & \mathsf{f}^\sharp(\;0\;,\;1\;,z,z,z) \\ \mathsf{a} & \to & 0 \\ \mathsf{a} & \to & 1 \end{array}$$

backward narrowing fails

$$\begin{aligned} \mathsf{f}^\sharp(x,x) &\to \mathsf{f}^\sharp(\mathsf{c}(2),\mathsf{c}(3)) \\ 2 &\to \mathsf{b} \\ 3 &\to \mathsf{b} \end{aligned}$$

narrowing fails

$$\begin{aligned} \mathsf{g}^\sharp(\mathsf{c}(0),\mathsf{c}(1),x) &\to \mathsf{g}^\sharp(x,x,x) \\ \mathsf{a} &\to 0 \\ \mathsf{a} &\to 1 \end{aligned}$$

narrowing and backward narrowing fail

$$\begin{aligned} \mathsf{f}^\sharp(x,x,y) &\to \mathsf{f}^\sharp(\mathsf{c}(2(y,y)),\mathsf{c}(3(y,y)),y) \\ 2(\mathsf{c}(0),\mathsf{c}(1)) &\to \mathsf{b} \\ 3(\mathsf{c}(0),\mathsf{c}(1)) &\to \mathsf{b} \\ \mathsf{a} &\to 0 \\ \mathsf{a} &\to 1 \end{aligned}$$

overlap closures fail

$$\begin{aligned} \mathsf{f}^\sharp(x,y,x,y,z) &\to \mathsf{f}^\sharp(\mathsf{c}(0),\mathsf{c}(1),z,z,z) \\ \mathsf{a} &\to 0 \\ \mathsf{a} &\to 1 \end{aligned}$$

Even if narrowing into variables is allowed

Example (Non-looping non-terminating TRS)

$$\begin{aligned} \mathsf{f}(\mathsf{true}, x) &\rightarrow \mathsf{f}(\mathsf{nat}(x), \mathsf{s}(x)) \\ \mathsf{nat}(0) &\rightarrow \mathsf{true} \\ \mathsf{nat}(\mathsf{s}(x)) &\rightarrow \mathsf{nat} \end{aligned}$$

Infinite reduction:

$$\begin{array}{lcll} \mathsf{f}(\mathsf{true}, 0) & \rightarrow_{\mathcal{R}} & \mathsf{f}(\mathsf{nat}(0), \mathsf{s}(0)) & \rightarrow_{\mathcal{R}}^{1} \\ \mathsf{f}(\mathsf{true}, \mathsf{s}(0)) & \rightarrow_{\mathcal{R}} & \mathsf{f}(\mathsf{nat}(\mathsf{s}(0)), \mathsf{s}(\mathsf{s}(0))) & \rightarrow_{\mathcal{R}}^{2} \\ \mathsf{f}(\mathsf{true}, \mathsf{s}(\mathsf{s}(0))) & \rightarrow_{\mathcal{R}} & \mathsf{f}(\mathsf{nat}(\mathsf{s}(\mathsf{s}(0))), \mathsf{s}(\mathsf{s}(\mathsf{s}(0)))) & \rightarrow_{\mathcal{R}}^{3} \\ \ldots & & & \end{array}$$