On Applying Dependency Pairs to Process Verification

Jürgen Giesl
RWTH Aachen
Germany

Thomas Arts
Ericsson Telecom
Sweden

Contents:

1. Process verification $\rightarrow$ termination problem of CTRS
2. Prove termination of CTRS by *dependency pairs*
    - extension of dependency pairs to conditional TRSs
    - *narrowing*, *rewriting*, *instantiating* dependency pairs

Encode Verification Problem as CTRS

$$\begin{array}{l} m \leq \mathsf{length}(st) \to^* \mathsf{true}, \\ \mathsf{empty}(\mathsf{fst}(m, st)) \to^* \mathsf{false} \quad \mid \\ \quad \mathsf{process}(st, m) \to \\ \quad \mathsf{process}(\mathsf{map\_f}(\mathsf{self}, \mathsf{nil}) ++ \mathsf{rest}(m, st), m) \end{array}$$

$$\begin{array}{l} m \leq \mathsf{length}(st) \to^* \mathsf{false}, \\ \mathsf{empty}(\mathsf{fst}(m, \mathsf{map\_f}(\mathsf{self}, \mathsf{nil}) ++ st)) \to^* \mathsf{false} \quad \mid \\ \quad \mathsf{process}(st, m) \to \\ \quad \mathsf{process}(\mathsf{rest}(m, \mathsf{map\_f}(\mathsf{self}, \mathsf{nil}) ++ st), m) \end{array}$$

Verification problem $\equiv$ CTRS left-right decreasing

Transform CTRS into TRS

$$\begin{array}{rcl} \mathsf{process}(st, m) & \to & \mathsf{if}_1(st, m, m \leq \mathsf{length}(st)) \\ \mathsf{if}_1(st, m, \mathsf{true}) & \to & \mathsf{if}_2(st, m, \mathsf{empty}(\mathsf{fst}(m, st))) \\ \mathsf{if}_2(st, m, \mathsf{false}) & \to & \mathsf{process}(\mathsf{map\_f}(\mathsf{self}, \mathsf{nil}) ++ \mathsf{rest}(m, st), m) \\ \mathsf{if}_1(st, m, \mathsf{false}) & \to & \mathsf{if}_3(st, m, \mathsf{empty}(\mathsf{fst}(m, \mathsf{map\_f}(\mathsf{self}, \mathsf{nil}) ++ st))) \\ \mathsf{if}_3(st, m, \mathsf{false}) & \to & \mathsf{process}(\mathsf{rest}(m, \mathsf{map\_f}(\mathsf{self}, \mathsf{nil}) ++ st), m) \end{array}$$

Verification problem $\equiv$ TRS innermost terminating

$$
\begin{aligned}
\mathsf{process}(st, m) &\rightarrow \mathsf{if}_1(st, m, m \leq \mathsf{length}(st)) \\
\mathsf{if}_1(st, m, \mathsf{true}) &\rightarrow \mathsf{if}_2(st, m, \mathsf{empty}(\mathsf{fst}(m, st))) \\
\mathsf{if}_2(st, m, \mathsf{false}) &\rightarrow \mathsf{process}(\mathsf{map\_f}(\mathsf{self}, \mathsf{nil}) ++ \mathsf{rest}(m, st), m) \\
\mathsf{if}_1(st, m, \mathsf{false}) &\rightarrow \mathsf{if}_3(st, m, \mathsf{empty}(\mathsf{fst}(m, \mathsf{map\_f}(\mathsf{self}, \mathsf{nil}) ++ st))) \\
\mathsf{if}_3(st, m, \mathsf{false}) &\rightarrow \mathsf{process}(\mathsf{rest}(m, \mathsf{map\_f}(\mathsf{self}, \mathsf{nil}) ++ st), m)
\end{aligned}
$$

Defined Symbols: $\mathsf{process}, \mathsf{if}_1, \mathsf{if}_2, \mathsf{if}_3, \mathsf{empty}, \mathsf{fst}, \mathsf{map\_f}, ++, \ldots$
Constructors: $\mathsf{true}, \mathsf{false}, \mathsf{self}, \mathsf{nil}, \ldots$

Definition
If $f(s_1, \ldots, s_n) \rightarrow C[g(t_1, \ldots, t_m)]$ is a rule of $\mathcal{R}$ and g is a defined symbol, then

$$\langle \mathsf{F}(s_1, \ldots, s_n),\ \mathsf{G}(t_1, \ldots, t_m) \rangle$$

is a *dependency pair* of $\mathcal{R}$.

Dependency Pairs:

$\langle \mathsf{PROCESS}(st, m), \mathsf{IF}_1(st, m, m \leq \mathsf{length}(st)) \rangle$
$\langle \mathsf{IF}_1(st, m, \mathsf{true}), \mathsf{IF}_2(st, m, \mathsf{empty}(\mathsf{fst}(m, st))) \rangle$
$\langle \mathsf{IF}_2(st, m, \mathsf{false}), \mathsf{PROCESS}(\mathsf{map\_f}(\mathsf{self}, \mathsf{nil}) ++ \mathsf{rest}(m, st), m) \rangle$
$\langle \mathsf{IF}_1(st, m, \mathsf{false}), \mathsf{IF}_3(st, m, \mathsf{empty}(\mathsf{fst}(m, \mathsf{map\_f}(\mathsf{self}, \mathsf{nil}) ++ st))) \rangle$
$\langle \mathsf{IF}_3(st, m, \mathsf{false}), \mathsf{PROCESS}(\mathsf{rest}(m, \mathsf{map\_f}(\mathsf{self}, \mathsf{nil}) ++ st), m) \rangle$

$\langle \mathsf{PROCESS}(st, m), \mathsf{LE}(m, \mathsf{length}(st)) \rangle$
$\langle \mathsf{PROCESS}(st, m), \mathsf{LENGTH}(st) \rangle$

$\ldots$

Definition

$\langle s_1, t_1 \rangle\ \langle s_2, t_2 \rangle \ldots$ is an *innermost chain*,
if there exists a substitution σ such that

- $t_j\sigma \xrightarrow{\mathsf{i}}{}^{*} s_{j+1}\sigma$ for consecutive pairs $\langle s_j, t_j \rangle$ and $\langle s_{j+1}, t_{j+1} \rangle$
- all $s_j\sigma$ in normal form

$$\langle \mathsf{PROCESS}(st, m), \mathsf{IF}_1(st, m, m \leq \mathsf{length}(st)) \rangle \tag{1}$$
$$\langle \mathsf{IF}_1(st, m, \mathsf{true}), \mathsf{IF}_2(st, m, \mathsf{empty}(\mathsf{fst}(m, st))) \rangle \tag{2}$$
$$\langle \mathsf{IF}_2(st, m, \mathsf{false}), \mathsf{PROCESS}(\ldots ++\, \mathsf{rest}(m, st), m) \rangle \tag{3}$$
$$\langle \mathsf{IF}_1(st, m, \mathsf{false}), \mathsf{IF}_3(st, m, \mathsf{empty}(\ldots)) \rangle \tag{4}$$
$$\langle \mathsf{IF}_3(st, m, \mathsf{false}), \mathsf{PROCESS}(\mathsf{rest}(m, \ldots ++\, st), m) \rangle \tag{5}$$

Cycles:
$\mathcal{P}_1 = \{(1), (2), (3)\}$, $\mathcal{P}_2 = \{(1), (4), (5)\}$, $\mathcal{P}_3 = \{(1), \ldots, (5)\}, \ldots$

Innermost Termination Criterion

A TRS is innermost terminating iff there is no infinite innermost chain from any cycle.

Automation:

$$\langle s_1, t_1 \rangle \langle s_2, t_2 \rangle \langle s_3, t_3 \rangle \ldots \in \mathcal{P}$$

$$s_1\sigma \geq_{\mathcal{P}} t_1\sigma \geq_{\mathcal{P}} s_2\sigma \geq_{\mathcal{P}} t_2\sigma \geq_{\mathcal{P}} s_3\sigma \geq_{\mathcal{P}} t_3\sigma \geq_{\mathcal{P}} \ldots$$

Theorem

A TRS $\mathcal{R}$ is innermost terminating if for each cycle $\mathcal{P}$ there is a well-founded weakly monotonic quasi-ordering $\geq_{\mathcal{P}}$ closed under substitution such that

- $s \geq_{\mathcal{P}} t$ for all dependency pairs from $\mathcal{P}$
- $s >_{\mathcal{P}} t$ for at least one dependency pair from $\mathcal{P}$
- $l \geq_{\mathcal{P}} r$ for all rules usable in reductions of $t\sigma$

$$(\langle s, t \rangle \in \mathcal{P},\ \sigma \text{ normal})$$

Resulting Inequalities for $\mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3$:

$$\begin{array}{rcll}
\mathsf{PROCESS}(st, m) & \geq & \mathsf{IF}_1(st, m, m \leq \mathsf{length}(st)) & (1)\\
\mathsf{IF}_1(st, m, \mathsf{true}) & \geq & \mathsf{IF}_2(st, m, \mathsf{empty}(\mathsf{fst}(m, st))) & (2)\\
\mathsf{IF}_2(st, m, \mathsf{false}) & > & \mathsf{PROCESS}(\ldots +\!\!+\, \mathsf{rest}(m, st), m) & (3)\\
\mathsf{IF}_1(st, m, \mathsf{false}) & \geq & \mathsf{IF}_3(st, m, \mathsf{empty}(\ldots)) & (4)\\
\mathsf{IF}_3(st, m, \mathsf{false}) & > & \mathsf{PROCESS}(\mathsf{rest}(m, \ldots +\!\!+\, st), m) & (5)\\
l & \geq & r \quad \text{for all rules with } root(l) \in & \\
 & & \{\leq, \mathsf{empty}, \mathsf{fst}, +\!\!+, \mathsf{f}, \ldots\} &
\end{array}$$

But: not satisfied by (quasi-)simplification orderings

Narrowing Dependency Pairs

$$\begin{aligned} \mathsf{fst}(0, x) &\to \mathsf{nil} \\ \mathsf{fst}(\mathsf{s}(n), \mathsf{nil}) &\to \mathsf{nil} \\ \mathsf{fst}(\mathsf{s}(n), h :: t) &\to h :: \mathsf{fst}(n, t) \end{aligned}$$

$$\begin{aligned} \mathsf{empty}(\mathsf{nil}) &\to \mathsf{true} \\ \mathsf{empty}(h :: t) &\to \mathsf{false} \end{aligned}$$

$$\begin{aligned} 0 \leq m &\to \mathsf{true} \\ \mathsf{s}(n) \leq 0 &\to \mathsf{false} \\ \mathsf{s}(n) \leq \mathsf{s}(m) &\to n \leq m \end{aligned}$$

$$\langle \mathsf{P}(st, m), \; \mathsf{IF}_1(st, m, m \leq \mathsf{len}(st)) \rangle$$

$$\langle \mathsf{IF}_1(st, m, \mathsf{true}), \; \mathsf{IF}_2(st, m, \mathsf{empty}(\mathsf{fst}(m, st))) \rangle$$

$$\langle \mathsf{IF}_2(st, m, \mathsf{false}), \; \mathsf{P}(\mathsf{map\_f}(\mathsf{self}, \mathsf{nil}) +\!\!+ \; \mathsf{rest}(m, st), m) \rangle$$

$$\langle \mathsf{IF}_1(x, 0, \mathsf{true}),\ \mathsf{IF}_2(x, 0, \mathsf{empty}(\mathsf{nil}))\rangle \qquad \langle \mathsf{IF}_1(\mathsf{nil}, \mathsf{s}(n), \mathsf{true}),\ \mathsf{IF}_2(\mathsf{nil}, \mathsf{s}(n), \mathsf{empty}(\mathsf{nil}))\rangle \qquad \boxed{\langle \mathsf{IF}_1(h :: t, \mathsf{s}(n), \mathsf{true}),\ \mathsf{IF}_2(h :: t, \mathsf{s}(n), \mathsf{empty}(h :: \mathsf{fst}(n, t)))\rangle}$$

$$\langle \mathsf{IF}_1(x, 0, \mathsf{true}),\ \mathsf{IF}_2(x, 0, \mathsf{true})\rangle \qquad \langle \mathsf{IF}_1(\mathsf{nil}, \mathsf{s}(n), \mathsf{true}),\ \mathsf{IF}_2(\mathsf{nil}, \mathsf{s}(n), \mathsf{true})\rangle$$

Narrowing Dependency Pairs

$$\begin{aligned}\mathsf{fst}(0, x) &\to \mathsf{nil}\\ \mathsf{fst}(\mathsf{s}(n), \mathsf{nil}) &\to \mathsf{nil}\\ \mathsf{fst}(\mathsf{s}(n), h :: t) &\to h :: \mathsf{fst}(n, t)\end{aligned}$$

$$\begin{aligned}\mathsf{empty}(\mathsf{nil}) &\to \mathsf{true}\\ \mathsf{empty}(h :: t) &\to \mathsf{false}\end{aligned}$$

$$\begin{aligned}0 \leq m &\to \mathsf{true}\\ \mathsf{s}(n) \leq 0 &\to \mathsf{false}\\ \mathsf{s}(n) \leq \mathsf{s}(m) &\to n \leq m\end{aligned}$$

$\langle \mathsf{P}(st, m),$
$\mathsf{IF}_1(st, m, m \leq \mathsf{len}(st))\rangle$

$\boxed{\begin{array}{c}\langle \mathsf{IF}_1(h :: t, \mathsf{s}(n), \mathsf{true}),\\ \mathsf{IF}_2(h :: t, \mathsf{s}(n), \ldots)\rangle\end{array}}$

$\langle \mathsf{IF}_2(st, m, \mathsf{false}),$
$\mathsf{P}(\mathsf{map\_f}(\mathsf{self}, \mathsf{nil}) +\!\!+$
$\mathsf{rest}(m, st), m)\rangle$

$\langle \mathsf{P}(\mathsf{nil}, m),$
$\mathsf{IF}_1(\mathsf{nil}, m, m \leq \mathsf{0})\rangle$

$\langle \mathsf{P}(h :: t, m),$
$\mathsf{IF}_1(h :: t, m, m \leq \mathsf{s}(\mathsf{len}(t)))\rangle$

$\langle \mathsf{P}(st, \mathsf{0}),$
$\mathsf{IF}_1(st, \mathsf{0}, \mathsf{true})\rangle$

Narrowing Dependency Pairs

$$\begin{aligned} \mathsf{fst}(0, x) &\to \mathsf{nil} \\ \mathsf{fst}(\mathsf{s}(n), \mathsf{nil}) &\to \mathsf{nil} \\ \mathsf{fst}(\mathsf{s}(n), h :: t) &\to h :: \mathsf{fst}(n, t) \end{aligned}$$

$$\begin{aligned} \mathsf{empty}(\mathsf{nil}) &\to \mathsf{true} \\ \mathsf{empty}(h :: t) &\to \mathsf{false} \end{aligned}$$

$$\begin{aligned} 0 \le m &\to \mathsf{true} \\ \mathsf{s}(n) \le 0 &\to \mathsf{false} \\ \mathsf{s}(n) \le \mathsf{s}(m) &\to n \le m \end{aligned}$$

$\langle \mathsf{P}(h :: t, m),$
$\mathsf{IF}_1(h :: t, m, \ldots)\rangle$

$\langle \mathsf{IF}_1(h :: t, \mathsf{s}(n), \mathsf{true}),$
$\mathsf{IF}_2(h :: t, \mathsf{s}(n), \ldots)\rangle$

$\langle \mathsf{IF}_2(st, m, \mathsf{false}),$
$\mathsf{P}(\mathsf{map\_f}(\mathsf{self}, \mathsf{nil}) +\!\!+$
$\mathsf{rest}(m, st), m)\rangle$

$\ldots \quad \ldots \quad \ldots$

$\langle \mathsf{IF}_2(h :: t, \mathsf{s}(n), \mathsf{false}),$
$\mathsf{P}(\mathsf{map\_f}(\mathsf{self}, \mathsf{nil}) +\!\!+ \mathsf{rest}(n, t), \mathsf{s}(n)) \rangle$

$\ldots \quad \ldots \quad \ldots$

$\langle \mathsf{IF}_2(h :: t, \mathsf{s}(n), \mathsf{false}),$
$\mathsf{P}(\mathsf{nil} +\!\!+ \mathsf{rest}(n, t), \mathsf{s}(n)) \rangle$

$\ldots \quad \ldots$

$\langle \mathsf{IF}_2(h :: t, \mathsf{s}(n), \mathsf{false}),$
$\mathsf{P}(\mathsf{rest}(n, t), \mathsf{s}(n)) \rangle$

Narrowing Dependency Pairs

$$\begin{aligned} \mathsf{fst}(0, x) &\to \mathsf{nil} \\ \mathsf{fst}(\mathsf{s}(n), \mathsf{nil}) &\to \mathsf{nil} \\ \mathsf{fst}(\mathsf{s}(n), h :: t) &\to h :: \mathsf{fst}(n, t) \end{aligned}$$

$$\begin{aligned} \mathsf{empty}(\mathsf{nil}) &\to \mathsf{true} \\ \mathsf{empty}(h :: t) &\to \mathsf{false} \end{aligned}$$

$$\begin{aligned} 0 \le m &\to \mathsf{true} \\ \mathsf{s}(n) \le 0 &\to \mathsf{false} \\ \mathsf{s}(n) \le \mathsf{s}(m) &\to n \le m \end{aligned}$$

$\langle \mathsf{P}(h :: t, m),$
$\mathsf{IF}_1(h :: t, m, \ldots)\rangle$

$\langle \mathsf{IF}_1(h :: t, \mathsf{s}(n), \mathsf{true}),$
$\mathsf{IF}_2(h :: t, \mathsf{s}(n), \ldots)\rangle$

$\langle \mathsf{IF}_2(h :: t, \mathsf{s}(n), \mathsf{false}),$
$\mathsf{P}(\mathsf{map\_f}(\mathsf{self}, \mathsf{nil}) +\!\!+ \mathsf{rest}(n, t), \mathsf{s}(n))\rangle$

$\langle \mathsf{IF}_2(h :: t, \mathsf{s}(n), \mathsf{false}),$
$\mathsf{P}(\mathsf{nil} +\!\!+ \mathsf{rest}(n, t), \mathsf{s}(n))\rangle$

$\langle \mathsf{IF}_2(h :: t, \mathsf{s}(n), \mathsf{false}),$
$\mathsf{P}(\mathsf{rest}(n, t), \mathsf{s}(n))\rangle$

$\langle s, t\rangle$ *narrows* to $\langle s\mu, t'\rangle$ iff t narrows to t' via μ.

Theorem (Narrowing Dependency Pairs)
TRS non-overlapping, $\langle s, t\rangle \in P$, t *does not unify* with any lhs in P, $P' := P$ with $\langle s, t\rangle$ replaced by *all* its narrowings.

No infinite innermost chain of pairs from P'	$\Leftrightarrow$	No infinite innermost chain of pairs from P

Theorem (Rewriting Dependency Pairs)
TRS non-overlapping, P results from dependency pairs, $\langle s, t\rangle \in P$, $t \to_{\mathcal{R}} r$, $P' := P$ with $\langle s, t\rangle$ replaced by $\langle s, r\rangle$

No infinite innermost chain of pairs from P'	$\Leftrightarrow$	No infinite innermost chain of pairs from P

$$\mathsf{P}(h :: t, m) \;\geq\; \mathsf{IF}_1(h :: t, m, m \leq \mathsf{s}(\mathsf{length}(t))) \qquad (1)$$

$$\mathsf{IF}_1(h :: t, \mathsf{s}(n), \mathsf{true}) \;\geq\; \mathsf{IF}_2(h :: t, \mathsf{s}(n), \mathsf{false}) \qquad (2)$$

$$\mathsf{IF}_2(h :: t, \mathsf{s}(n), \mathsf{false}) \;>\; \mathsf{P}(\mathsf{rest}(n, t), \mathsf{s}(n)) \qquad (3.3)$$

$$\mathsf{IF}_1(h :: t, \mathsf{s}(n), \mathsf{false}) \;\geq\; \mathsf{IF}_3(h :: t, \mathsf{s}(n), \mathsf{false}) \qquad (4)$$

$$\mathsf{IF}_3(h :: t, \mathsf{s}(n), \mathsf{false}) \;>\; \mathsf{P}(\mathsf{rest}(n, t), \mathsf{s}(n)) \qquad (5)$$

$$\begin{aligned} \mathsf{rest}(n, x) &\;\geq\; x \\ \mathsf{rest}(\mathsf{s}(n), x) &\;\geq\; \mathsf{nil} \\ \mathsf{rest}(\mathsf{s}(n), h :: t) &\;\geq\; \mathsf{rest}(n, t) \end{aligned}$$

$$l \;\geq\; r \;\; \text{for all rules with } root(l) \in \{\leq, \mathsf{len}\}$$

Can be proved using the recursive path ordering

$\Rightarrow$ TRS innermost terminating, process property verified!

Verification Problem: if first process receives empty message,
then last also sends empty message

Encode Verification Problem as CTRS:

$$\mathsf{ring}(st_1, in_2, st_2, in_3, st_3, m) \to \mathsf{ring}(\mathsf{rest}(m, st_1), \ldots) \mid \mathsf{empty}(\mathsf{fst}(m, st_1)) \to^* \mathsf{false}$$

$$\vdots$$

Verification problem $\equiv$ CTRS *left-right decreasing*

Transform CTRS into TRS:

$$\begin{aligned}
\mathsf{ring}(st_1, in_2, st_2, in_3, st_3, m) &\to \mathsf{if}_1(\ldots, \mathsf{empty}(\mathsf{fst}(m, st_1))) \\
\mathsf{if}_1(\ldots, \mathsf{false}) &\to \mathsf{ring}(\mathsf{rest}(n, st_1), \ldots) \\
&\vdots \\
\mathsf{ring}(st_1, in_2, st_2, in_3, st_3, m) &\to \mathsf{if}(\ldots) \\
\mathsf{if}(\ldots) &\to \mathsf{ring}(\ldots) \\
&\vdots
\end{aligned}$$

Verification problem $\equiv$ TRS *terminating*

Networks of Asynchronous Processes

results in *overlapping* TRS (many rules $\mathsf{ring}(\ldots) \to \ldots$)

Problem: innermost termination $\not\Rightarrow$ termination

Theorem

If $\mathcal{R} = \mathcal{R}_1 \cup \mathcal{R}_2$ with

- $\mathcal{R}_1$ non-overlapping $\leftarrow$ auxiliary rules
- $\mathcal{R}_2$ non-collapsing $\leftarrow$ process network rules
- no critical pairs between $\mathcal{R}_1$ and $\mathcal{R}_2$
- roots of lhs & rhs of $\mathcal{R}_2$ are not in $\mathcal{R}_1$ and not below root in $\mathcal{R}_2$

then innermost termination of $\mathcal{R}$ implies termination of $\mathcal{R}$.

$\Rightarrow$ innermost termination enough for asynchronous networks

Instantiating Dependency Pairs

$$\begin{aligned} \mathsf{fst}(0, x) &\to \mathsf{nil} \\ \mathsf{fst}(\mathsf{s}(n), \mathsf{nil}) &\to \mathsf{nil} \\ \mathsf{fst}(\mathsf{s}(n), h :: t) &\to h :: \mathsf{fst}(n, t) \end{aligned} \qquad \begin{aligned} \mathsf{empty}(\mathsf{nil}) &\to \mathsf{true} \\ \mathsf{empty}(h :: t) &\to \mathsf{false} \end{aligned} \qquad \begin{aligned} \mathsf{rest}(n, x) &\to x \\ \mathsf{rest}(\mathsf{s}(n), x) &\to \mathsf{nil} \\ \mathsf{rest}(\mathsf{s}(n), h :: t) &\to \mathsf{rest}(n, t) \end{aligned}$$

$$\begin{aligned} &\langle \mathsf{RING}(st_1, \ldots, m), \\ &\ \mathsf{IF}_1(st_1, \ldots, m, \mathsf{empty}(\mathsf{fst}(m, st_1)))\rangle \end{aligned} \quad (1)$$

$$\begin{aligned} &\langle \mathsf{IF}_1(st_1, \ldots, m, \mathsf{false}), \\ &\ \mathsf{RING}(\mathsf{rest}(m, st_1), \ldots, m)\rangle \end{aligned} \quad (2)$$

Narrowing &
Rewriting

$$\boxed{\begin{aligned} &\langle \mathsf{RING}(h :: t, \ldots, \mathsf{s}(n)), \\ &\ \mathsf{IF}_1(h :: t, \ldots, \mathsf{s}(n), \mathsf{false})\rangle \end{aligned}} \quad (1')$$

Narrowing of $\langle s, t\rangle$ only allowed if t *does not unify* with any lhs.

But there are dependency pairs:
$\langle\, \mathsf{RING}(st_1, \ldots, m),\ \mathsf{IF}_2(\ldots)\,\rangle$!

$\langle \mathsf{RING}(h :: t, \ldots, \mathsf{s}(n)),$
$\mathsf{IF}_1(h :: t, \ldots, \mathsf{s}(n), \mathsf{false})\rangle$

$(1')$

$\langle \mathsf{IF}_1(st_1, \ldots, m, \mathsf{false}),$
$\mathsf{RING}(\mathsf{rest}(m, st_1), \ldots, m)\rangle$ (2)

Instantiation

$\langle \mathsf{IF}_1(h :: t, \ldots, \mathsf{s}(n), \mathsf{false}),$
$\mathsf{RING}(\mathsf{rest}(\mathsf{s}(n), h :: t), \ldots, \mathsf{s}(n))\rangle$

Theorem

P set of pairs, $\langle v, w\rangle, \langle s, t\rangle, \in P$, $P' := P$ with $\langle s, t\rangle$ replaced by all $\langle s\mu, t\mu\rangle$ where $\mu \in \mathrm{mgu}(\textsc{cap}(w), s)$.

No infinite innermost chain of pairs from P'	$\Leftrightarrow$	No infinite innermost chain of pairs from P

Example: After narrowing, rewriting, instantiating DPs, constraints satisfied by recursive path ordering

$\Rightarrow$ TRS is (innermost) terminating, network property verified!

Conclusion

- Dependency Pairs applicable for process verification in industry
  - process verification $\rightarrow$

    termination problem of CTRS $\rightarrow$

    innermost termination problem of TRS
  - dependency pairs for automated innermost termination proof
- Several new results developed
  - dependency pairs for conditional TRSs
  - narrowing, rewriting, instantiating dependency pairs
  - innermost termination enough for asynchronous processes