CO3
a COnverter for proving COfluence of COnditional TRSs (Version 1.2)

Naoki Nishida[1], Takayuki Kuroda[1], Makishi Yanagisawa[1] and Karl Gmeiner[2]

[1] Nagoya University, Nagoya, Japan
{nishida@, kuroda@apal.i., makishi@trs.cm.}is.nagoya-u.ac.jp
[2] UAS Technikum Wien, Vienna, Austria
gmeiner@technikum-wien.at

CO3 is a tool for proving confluence of conditional term rewriting systems (CTRS) by using a transformational approach. The tool is based on the result in [4]: the tool first transforms a given normal 1-CTRS into an unconditional term rewriting system (TRS) by using the *SR transformation* [6] or the *unraveling* [3, 5], and then verify confluence of the transformed TRS. This tool is basically a converter of CTRSs to TRSs. The main expected use of this tool is the collaboration with other tools for proving confluence of TRSs, and thus this tool has very simple and lightweight functions to verify properties such as confluence and termination of TRSs. The tool is available from `http://www.trs.cm.is.nagoya-u.ac.jp/co3/` via a web interface.

The tool supports *normal 1-CTRSs* without any strategy and theory (specified by `STRATEGY` and `THEORY`, resp.), the class of which includes *TRSs*. Due to a technical reason as shown below, the tool is working for *weakly left-linear* CTRSs which are not TRSs. To enter the competition, the scope of the tool was modified to *oriented 1-CTRSs*.

The main technique in this tool is based on the following theorem: a weakly left-linear normal 1-CTRS $\mathcal{R}$ is confluent if one of $\mathbb{SR}(\mathcal{R})$ and $\mathbb{U}(\mathcal{R})$ is confluent [4], where the (optimized) SR transformation [6] and the sequential (optimized) *unraveling* are denoted by $\mathbb{SR}$ and $\mathbb{U}$, resp. For proving confluence and termination of TRSs, CO3 is using the following very fundamental (sufficient) conditions: (Confluence) "orthogonality" and "termination and joinability of all the critical pairs"; (Termination) "non-existence of SCCs in the *estimated dependency graph* [1]" and "the *dependency pair theorem* [1, Theorem 7] with the *reduction order* based on term-size and variable-occurrence [2, Example 5.2.2]".

The main new feature for CoCo 2015 is to drop *infeasible* rewrite rules. Implemented sufficient conditions for infeasibility are (1) "non-unifiability for the both sides of conditions under $REN(CAP(\cdot))$ in [1]", (2) "left-to-right unreachability of conditions at the root position", and (3) "trivial divergence of evaluating conditions".

References

[1] T. Arts and J. Giesl. Termination of term rewriting using dependency pairs. *Theor. Comput. Sci.*, 236(1-2):133–178, 2000.

[2] F. Baader and T. Nipkow. *Term Rewriting and All That.* Cambridge University Press, 1998.

[3] M. Marchiori. Unravelings and ultra-properties. In *Proc. ALP 1996*, vol. 1139 of *LNCS*, pp. 107–121. Springer, 1996.

[4] N. Nishida, M. Yanagisawa, and K. Gmeiner. On proving confluence of conditional term rewriting systems via the computationally equivalent transformation. In *Proc. IWC 2014*, pp. 24–28, 2014.

[5] E. Ohlebusch. Termination of logic programs: Transformational methods revisited. *Appl. Algebra Eng. Commun. Comput.*, 12(1/2):73–116, 2001.

[6] T.-F. Şerbănuţă and G. Roşu. Computationally equivalent elimination of conditions. In *Proc. RTA 2006*, vol. 4098 of *LNCS*, pp. 19–34. Springer, 2006.

A Implemented Procedure

Given a CTRS $\mathcal{R}$, the tool performs as follows:

1. If $\mathcal{R}$ is a normal 1-CTRS,[1] then go to the next step, and otherwise, stop with printing `UNSUPPORTED`.

2. Transform a conditional rule

   $$l \to r \Leftarrow s_1 \twoheadrightarrow t_1, \ldots, s_{i-1} \twoheadrightarrow t_{i-1}, c(u_1, \ldots, u_n) \twoheadrightarrow c(v_1, \ldots, v_n), s_{i+1} \twoheadrightarrow t_{i+1}, \ldots, s_k \twoheadrightarrow t_k$$

   into

   $$l \to r \Leftarrow s_1 \twoheadrightarrow t_1, \ldots, s_{i-1} \twoheadrightarrow t_{i-1}, u_1 \twoheadrightarrow v_1, \ldots, u_n \twoheadrightarrow v_n, s_{i+1} \twoheadrightarrow t_{i+1}, \ldots, s_k \twoheadrightarrow t_k$$

   as much as possible, where c is a constructor.

3. $\mathcal{R} := \{l \to r \Leftarrow c \in \mathcal{R} \mid$ it is succeeded in proving that $l \to r \Leftarrow c$ is infeasible in $\mathcal{R}\}$. Infeasibility of $l \to r \Leftarrow s_1 \twoheadrightarrow t_1, \ldots, s_k \twoheadrightarrow t_k$ is examined by the following sufficient conditions in order:

   (a) There exists some $i \in \{1, \ldots, k\}$ such that $REN(CAP(s_i))$ and t_i are not unifiable.

   (b) There exists some $i \in \{1, \ldots, k\}$ such that t_i is not a variable, $root(t_i) \notin F$, where F is the smallest set of symbols, which satisfies all the following conditions:

   - $root(s_i) \in F$, and
   - if $l' \to r' \Leftarrow c' \in \mathcal{R}$ and $root(l') \in F$, then
     - if r' is a variable, then all symbols are in F, and
     - otherwise, $root(r') \in F$.

4. $\mathcal{R} := \mathcal{R} \setminus (\bigcup_{f \text{ is defined by } \mathcal{R}} \mathcal{R}_f)$ where $\mathcal{R}_f$ is the set of rules defining f and each f-rule in $\mathcal{R}_f$ has a condition of the form $f(u_1, \ldots, u_n) \twoheadrightarrow t$ such that $REN(f(CAP(u_1), \ldots, CAP(u_n)))$ and t are not unifiable.[2]

5. Try to disprove confluence by the following simple criterion: there exists an *unconditional* critical pair (s, t) of $\mathcal{R}$ such that s and t are different ground normal forms[3] w.r.t. $\mathcal{R}$. If this criterion is satisfied, then stop with printing `NO`, and otherwise, go to the next step.

6. If $\mathcal{R}$ is a TRS, then let $\mathcal{R}' := \mathcal{R}$ and go to Step 9.

7. If $\mathcal{R}$ is weakly left-linear, then go to the next step, and otherwise, stop with printing `MAYBE`.

8. Apply $\mathbb{SR}$ and $\mathbb{U}$ to $\mathcal{R}$, obtaining $\mathcal{R}'$ and $\mathcal{R}''$ by $\mathcal{R}' := \mathbb{SR}(\mathcal{R})$ and $\mathcal{R}'' := \mathbb{U}(\mathcal{R})$. Then, go to the next step. Note that in applying $\mathbb{SR}$ to $\mathcal{R}$, if $\mathcal{R}$ is a constructor system, then we do not introduce a special unary symbol wrapping the evaluation of conditions, that is, we apply the transformation of Antoy, Brassel, and Hanus [1] (see [6]).

[1] Note that TRSs are normal 1-CTRSs.

[2] This approach can be extended to the set of *mutually* defined functions for f.

[3] To examine whether a term is a ground normal form, we use a simple decidable sufficient condition that the term is a ground normal form w.r.t. $\mathcal{R}_u := \{l \to r \mid l \to r \Leftarrow c \in \mathcal{R}\}$.

9. Verify confluence of $\mathcal{R}'$ and $\mathcal{R}''$ by using the existing criteria (shown below) or other tools for proving confluence of TRSs.[4] If $\mathcal{R}'$ is confluent, then stop with printing `YES`, and otherwise, stop with printing `MAYBE`. To prove confluence, the current version of the tool is using the following criteria in order:

   (a) $\mathcal{R}'$ is orthogonal.

   (b) $\mathcal{R}'$ is terminating[5] and all the critical pairs of $\mathcal{R}'$ are joinable. If $\mathcal{R}$ is a TRS and there exists a non-joinable critical pair, then stop with printing `NO`.

To make the effect of the improvement clear, we show two examples.

Example 1. *Consider the following oriented CTRS:*

$$\mathcal{R} = \left\{ \begin{array}{rcl} \mathsf{f}(\mathsf{c}(x), \mathsf{c}(\mathsf{c}(y))) \to \mathsf{a}(\mathsf{a}(x)) & \Leftarrow & \mathsf{c}(\mathsf{f}(x,y)) \twoheadrightarrow \mathsf{c}(\mathsf{a}(\mathsf{b})) \\ \mathsf{f}(\mathsf{c}(\mathsf{c}(\mathsf{c}(x))), y) \to \mathsf{a}(y) & \Leftarrow & \mathsf{c}(\mathsf{f}(\mathsf{c}(x), \mathsf{c}(\mathsf{c}(y)))) \twoheadrightarrow \mathsf{c}(\mathsf{a}(\mathsf{a}(\mathsf{b}))) \\ \mathsf{h}(\mathsf{b}) \to \mathsf{b} & & \\ \mathsf{h}(\mathsf{a}(\mathsf{a}(x))) \to \mathsf{a}(\mathsf{b}) & \Leftarrow & \mathsf{h}(\mathsf{x}) \twoheadrightarrow \mathsf{b} \end{array} \right\}$$

CO3 converts this CTRS into

$$\mathcal{R}' = \left\{ \begin{array}{rcl} \mathsf{f}(\mathsf{c}(x), \mathsf{c}(\mathsf{c}(y))) \to \mathsf{a}(\mathsf{a}(x)) & \Leftarrow & \mathsf{f}(x,y) \twoheadrightarrow \mathsf{a}(\mathsf{b}) \\ \mathsf{f}(\mathsf{c}(\mathsf{c}(\mathsf{c}(x))), y) \to \mathsf{a}(y) & \Leftarrow & \mathsf{f}(\mathsf{c}(x), \mathsf{c}(\mathsf{c}(y))) \twoheadrightarrow \mathsf{a}(\mathsf{a}(\mathsf{b})) \\ \mathsf{h}(\mathsf{b}) \to \mathsf{b} & & \\ \mathsf{h}(\mathsf{a}(\mathsf{a}(x))) \to \mathsf{a}(\mathsf{b}) & \Leftarrow & \mathsf{h}(\mathsf{x}) \twoheadrightarrow \mathsf{b} \end{array} \right\}$$

and then into

$$\mathcal{R}'' = \{ \ \mathsf{h}(\mathsf{b}) \to \mathsf{b}, \quad \mathsf{h}(\mathsf{a}(\mathsf{a}(x))) \to \mathsf{a}(\mathsf{b})\mathsf{h}(\mathsf{x}) \twoheadrightarrow \mathsf{b} \Leftarrow \quad \}$$

$\mathcal{R}''$ is orthogonal, and then CO3 suceeds in proving confluence of $\mathcal{R}''$ and thus $\mathcal{R}$.

Example 2. *Consider the following oriented CTRS:*

$$\mathcal{R} = \left\{ \begin{array}{rcl} \mathsf{f}(\mathsf{c}(x), \mathsf{c}(\mathsf{c}(y))) \to \mathsf{a}(\mathsf{a}(x)) & \Leftarrow & \mathsf{c}(\mathsf{f}(x,y)) \twoheadrightarrow \mathsf{c}(\mathsf{a}(\mathsf{b})) \\ \mathsf{f}(\mathsf{c}(\mathsf{c}(\mathsf{c}(x))), y) \to \mathsf{a}(y) & \Leftarrow & \mathsf{c}(\mathsf{f}(\mathsf{c}(x), \mathsf{c}(\mathsf{c}(y)))) \twoheadrightarrow \mathsf{c}(\mathsf{a}(\mathsf{a}(\mathsf{b}))) \\ \mathsf{a}(x) \to \mathsf{b} & & \\ \mathsf{a}(x) \to \mathsf{f}(x,x) & & \end{array} \right\}$$

As in Example 1, CO3 converts this CTRS into

$$\mathcal{R}' = \{ \ \mathsf{a}(x) \to \mathsf{b}, \quad \mathsf{a}(x) \to \mathsf{f}(x,x) \ \}$$

This TRS clearly has a critical peak which is not joinable: $\mathsf{b} \leftarrow_{\mathcal{R}''} \mathsf{a}(x) \to_{\mathcal{R}''} \mathsf{f}(x,x)$. Thus, $\mathcal{R}'$ is not confluence, and CO3 suceeds in disproving confluence of $\mathcal{R}$.

References

[1] S. Antoy, B. Brassel, and M. Hanus. Conditional narrowing without conditions. In *Proc. PPDP 2003*, pp. 20–31. ACM, 2003.

[4] When $\mathcal{R}' = \mathcal{R}$, we do not consider the case of $\mathcal{R}''$.

[5] To prove termination, we use the following two sufficient conditions in order:

- non-existence of SCCs in the estimated dependency graph, and
- the dependency pair theorem (i.e., $(\forall l \to r \in \mathcal{R}'.\ l \succsim r) \wedge (\forall s \to t \in DP(\mathcal{R}').\ s \succ t)$) with the reduction pair $(\succ, \succsim)$ where $\succ := \{(u,v) \mid |u| > |v|,\ \forall x \in \mathcal{V}.\ |u|_x \geq |v|_x\}$ and $\succsim := \{(u,v) \mid |u| \geq |v|,\ \forall x \in \mathcal{V}.\ |u|_x \geq |v|_x\}$ (see [2, Example 5.2.2]).