# CO3: a COnverter for proving COfluence of COnditional term rewriting systems*

Naoki Nishida
Nagoya University
nishida@is.nagoya-u.ac.jp

Makishi Yanagisawa
Nagoya University
makishi@trs.cm.is.nagoya-u.ac.jp

Karl Gmeiner
UAS Technikum Wien
gmeiner@technikum-wien.at

June 27, 2014

## 1 Overview

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* [5], and then verify confluence of the transformed TRS.

The tool is available from `http://www.trs.cm.is.nagoya-u.ac.jp/co3/` via a web interface.

## 2 Supported Classes

The tool supports *normal 1-CTRSs* without any strategy and theory (specified by `STRATEGY` and `THEORY`, resp.), the class of which includes *TRSs*. The implemented techniques for TRSs are very poor since the tool is focusing on CTRSs. Due to a technical reason as shown later, indeed the tool is working for *weakly left-linear* CTRSs which has at least one condition.

## 3 Technical Background

The (optimized) SR transformation [5] is denoted by $\mathbb{SR}$.

**Theorem 1** *A weakly left-linear normal 1-CTRS $\mathcal{R}$ is confluent if $\mathbb{SR}(\mathcal{R})$ is confluent [4]. An orthogonal TRS is confluent [2]. A terminating TRS is confluent if all its critical pairs are joinable [3].*

## 4 Implemented Procedure

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

1. If $\mathcal{R}$ is a normal 1-CTRS, then go to the next step, and otherwise, stop with printing "`UNSUPPORTED`".
2. If $\mathcal{R}$ is a TRS, then let $\mathcal{R}' := \mathcal{R}$ and go to Step 5.
3. If $\mathcal{R}$ is weakly left-linear, then go to the next step, and otherwise, stop with printing "`MAYBE`".[1]
4. Apply $\mathbb{SR}$ to $\mathcal{R}$, obtaining $\mathcal{R}'$ by $\mathcal{R}' := \mathbb{SR}(\mathcal{R})$. Then, go to the next step. Note that 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 [5]).
5. Verify confluence of $\mathcal{R}'$ by using the existing criteria or other tools for proving confluence of TRSs. If $\mathcal{R}'$ is confluence, then stop with printing "`YES`", and otherwise, go to the next step. The current version of the tool is checking whether (i) $\mathcal{R}'$ is orthogonal, or (ii) $\mathcal{R}'$ is terminating (under a very simple syntactic criterion) and all critical pairs of $\mathcal{R}'$ are joinable.
6. 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 and strongly irreducible w.r.t. $\mathcal{R}$. If this criterion is satisfied, then stop with printing "`NO`", and otherwise, stop with printing "`MAYBE`".

One can use the tool by the following command:

`co3` *trs-format-file*

The output of the execution is "`UNSUPPORTED`", "`YES`", "`NO`", or "`MAYBE`" (or an error message if unexpected errors (e.g., on syntax) happen).

## 5 Remarks

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. For this purpose, the tool provides a pure converter, named CO2 (a COnverter of COnditional term rewriting systems):

`co2` *trs-format-file*

The result is a TRS in the `trs` format, and nothing if the input is unsupported.

## References

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

[2] J. W. Klop. *Term Rewriting Systems*, volume 2 of *Handbook of Logic in Computer Science*, pages 1–116. Oxford University Press, 1992.

[3] D. E. Knuth and P. B. Bendix. Simple word problems in universal algebras. *Computational problems in abstract algebra*, pages 263–297, 1970.

[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*, 2014. to appear.

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

*The research in this paper is partly supported by the Austrian Science Fund (FWF) international project I963 and the Japan Society for the Promotion of Science (JSPS).

[1] We may use criteria for proving confluence of CTRSs directly, but we did not implement such an approach in order to concentrate on the technique using transformations.