CO3 (Version 1.5)

Naoki Nishida[1,†], Yuta Tsuruta[1], and Yoshiaki Kanazawa[1,‡]

Nagoya University, Nagoya, Japan
{†nishida@, ‡yoshiaki@trs.css.}i.nagoya-u.ac.jp

CO3, a converter for proving confluence of conditional TRSs, tries to prove confluence of conditional term rewriting systems (CTRSs, for short) by using a transformational approach. The tool is based on the result in [3, 8, 6]: the tool first transforms a given weakly-left-linear (WLL, for short) 3-DCTRS into an unconditional term rewriting system (TRS, for short) by using the *SR transformation* $\mathbb{SR}$ [10, 11, 5] or the *unravelings* $\mathbb{U}_N$ [4] and $\mathbb{U}$ [9], and then verifies confluence of the transformed TRS by using the following theorems: (a) a normal 1-CTRS $\mathcal{R}$ is confluent if $\mathcal{R}$ is WLL and $\mathbb{U}_N(\mathcal{R})$ or $\mathbb{U}(\mathcal{R})$ is confluent [1, 2, 3], (b) a 3-DCTRS $\mathcal{R}$ is confluent if $\mathcal{R}$ is WLL and $\mathbb{U}(\mathcal{R})$ is confluent [2, 3], (c) a WLL normal 1-CTRS $\mathcal{R}$ is confluent if $\mathbb{SR}(\mathcal{R})$ is confluent [8], and (d) a WLL and ultra-WLL 3-DCTRS $\mathcal{R}$ is confluent if $\mathbb{SR}(\mathcal{R})$ is confluent [6]. This tool is basically a converter of CTRSs to TRSs and the main expected use of this tool is the collaboration with other tools for proving confluence of TRSs. For this reason, 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.css.i.nagoya-u.ac.jp/co3/.

Since version 1.4, CO3 does not use $\mathbb{SR}$ because $\mathbb{SR}(\mathcal{R})$ is not confluent due to some auxiliary rules (see [6]), and the latest version does not use $\mathbb{U}_N$ because the power of proving confluence of normal 1-CTRSs by $\mathbb{U}_N$ is empirically weaker than that by $\mathbb{U}$ under the implemented sufficient conditions for confluence (see [7]).

References

[1] K. Gmeiner, B. Gramlich, and F. Schernhammer. On (un)soundness of unravelings. In *Proc. RTA 2010*, volume 6 of *LIPIcs*, pp. 119–134. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2010.

[2] K. Gmeiner, B. Gramlich, and F. Schernhammer. On soundness conditions for unraveling deterministic conditional rewrite systems. In *Proc. RTA 2012*, volume 15 of *LIPIcs*, pp. 193–208. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2012.

[3] K. Gmeiner, N. Nishida, and B. Gramlich. Proving confluence of conditional term rewriting systems via unravelings. In *Proc. IWC 2013*, pp. 35–39, 2013.

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

[5] R. Nakayama, N. Nishida, and M. Sakai. Sound structure-preserving transformation for ultra-weakly-left-linear deterministic conditional term rewriting systems. In *Proc. WPTE 2016*, volume 235 of *EPTCS*, pp. 62–77. Open Publishing Association, 2017.

[6] N. Nishida. Notes on confluence of ultra-WLL SDCTRSs via a structure-preserving transformation. In *Proc. IWC 2016*, pp. 60–64, 2016.

[7] N. Nishida, Y. Tsuruta, and Y. Kanazawa. Convergence of simultaneously and sequentially unraveled trss for normal conditional TRSs. In *Proc. IWC 2018*, pp. 57–62, 2018.

[8] 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.

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

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

[11] T.-F. Şerbănuţă and G. Roşu. Computationally equivalent elimination of conditions. Technical Report UIUCDCS-R-2006-2693, Department of Computer Science, University of Illinois at Urbana-Champaign, 2006.