CO3 (Version 2.7)

Naoki Nishida and Misaki Kojima

Nagoya University, Nagoya, Japan
nishida@i.nagoya-u.ac.jp kojima@i.nagoya-u.jp

CO3, a **CO**nverter for proving **CO**nfluence of **CO**nditional TRSs,[1] tries to prove confluence of conditional term rewrite systems (CTRSs, for short) by using a transformational approach (cf. [7]). The tool first transforms a given weakly-left-linear (WLL, for short) 3-DCTRS into an unconditional term rewrite system (TRS, for short) by using $\mathbb{U}_{conf}$ [2], a variant of the *unraveling* $\mathbb{U}$ [9], and then verifies confluence of the transformed TRS by using the following theorem.

Theorem 1 ([1, 2]). *A 3-DCTRS $\mathcal{R}$ is confluent if $\mathcal{R}$ is WLL and $\mathbb{U}_{conf}(\mathcal{R})$ is confluent.*

The tool is very efficient because of very simple and lightweight functions to verify properties such as confluence and termination of TRSs.

Since version 2.0, a *narrowing-tree*-based approach [8, 3] to prove infeasibility of a condition w.r.t. a CTRS has been implemented [4]. The approach is applicable to *syntactically deterministic* CTRSs that are operationally terminating and *ultra-right-linear* w.r.t. the *optimized* unraveling. To prove infeasibility of a condition c, the tool first proves confluence, and then linearizes c if failed to prove confluence; then, the tool computes and simplifies a narrowing tree for c, and examines the emptiness of the narrowing tree. Since version 2.2, CO3 accepts both *join* and *semi-equational* CTRSs, and transforms them into equivalent DCTRSs to prove confluence or infeasibility [5].

The difference from the previous version [6] is a light-weight implementation of LPO and RPO used in the reduction pair processor for termination. Since the main goal of CO3 is to provide a converter of CTRSs to TRSs, CO3 has maintained a design policy that does not rely on external tools. For this reason, CO3 does not call any SAT solver and uses the precedence determined as follows: Any defined symbol is greater than any constructor, and regarding symbols of the same type (defined symbols or constructors), those declared earlier are greater than those declared later. Unfortunately, the implementation of LPO and RPO has not yielded any new successful (dis)proofs regarding CTRSs in the ARI database, while confluence of `1498.ari` has newly been proved. For this reason, we register for the `TRS` category this year.

References

[1] K. Gmeiner, B. Gramlich, and F. Schernhammer. On soundness conditions for unraveling deterministic conditional rewrite systems. In A. Tiwari, editor, *Proceedings of the 23rd International Conference on Rewriting Techniques and Applications*, volume 15 of *Leibniz International Proceedings in Informatics*, pages 193–208. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2012.

[2] K. Gmeiner, N. Nishida, and B. Gramlich. Proving confluence of conditional term rewriting systems via unravelings. In N. Hirokawa and V. van Oostrom, editors, *Proceedings of the 2nd International Workshop on Confluence*, pages 35–39, 2013.

[3] Y. Maeda, N. Nishida, M. Sakai, and T. Kobayashi. Extending narrowing trees to basic narrowing in term rewriting. IEICE Technical Report SS2018-39, the Institute of Electronics, Information and Communication Engineers, 2019. Vol. 118, No. 385, pp. 73–78, in Japanese.

[1] http://www.lctrs.jp/tools/co3/

[4] N. Nishida. CO3 (Version 2.1). In M. Ayala-Rinćon and S. Mimram, editors, *Proceedings of the 9th International Workshop on Confluence*, page 67, 2020.

[5] N. Nishida. CO3 (Version 2.2). In S. Mimram and C. Rocha, editors, *Proceedings of the 10th International Workshop on Confluence*, page 61, 2021.

[6] N. Nishida and M. Kojima. CO3 (Version 2.6). In R. Gutiérrez and N. Nishida, editors, *Proceedings of the 14th International Workshop on Confluence*, pages 65–66, 2025.

[7] N. Nishida, T. Kuroda, and K. Gmeiner. CO3 (Version 1.3). In B. Accattoli and A. Tiwari, editors, *Proceedings of the 5th International Workshop on Confluence*, page 74, 2016.

[8] N. Nishida and Y. Maeda. Narrowing trees for syntactically deterministic conditional term rewriting systems. In H. Kirchner, editor, *Proceedings of the 3rd International Conference on Formal Structures for Computation and Deduction*, volume 108 of *Leibniz International Proceedings in Informatics*, pages 26:1–26:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018.

[9] E. Ohlebusch. Termination of logic programs: Transformational methods revisited. *Applicable Algebra in Engineering, Communication and Computing*, 12(1/2):73–116, 2001.