CO3 (Version 2.1)

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

CO3, a converter for proving confluence of conditional TRSs,[1] tries to prove confluence of conditional term rewriting systems (CTRSs, for short) by using a transformational approach (cf. [4]). 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 $\mathbb{U}_{conf}$ [2], a variant of the *unraveling* $\mathbb{U}$ [6], and then verifies confluence of the transformed TRS by using the following theorem: a 3-DCTRS $\mathcal{R}$ is confluent if $\mathcal{R}$ is WLL and $\mathbb{U}_{conf}(\mathcal{R})$ is confluent [1, 2]. 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 [5, 3] to prove infeasibility of a condition w.r.t. a specified CTRS has been implemented. The approach is applicable to *syntactically deterministic* CTRSs that are operationally terminating and *ultra-right-linear* w.r.t. the *optimized* unraveling. In the present version, bugs in version 2.0 has been fixed and the computation of SCCs for termination has slightly been improved.

To prove confluence by means of narrowing trees, the tool first computes the (conditional) critical pairs, and then proves their joinability as follows: a critical pair $\langle s,t\rangle \Leftarrow c$ is joinable if (1) c is the empty list and $s = t$, or (2) the narrowing tree for c can be simplified to a tree that defines the empty set of substitutions. For example, let us consider `489.trs` in Cops which is an operationally terminating normal 1-CTRS, and has a conditional critical pair $\langle \mathsf{true}, \mathsf{false}\rangle \Leftarrow \mathsf{o}(x) \twoheadrightarrow \mathsf{true}, \mathsf{e}(x) \twoheadrightarrow \mathsf{true}$. As a narrowing tree for condition $\mathsf{o}(x) \twoheadrightarrow \mathsf{true}, \mathsf{e}(x) \twoheadrightarrow \mathsf{true}$ w.r.t. `489.trs`, we construct the following production rules for a regular tree grammar [5]:

$$\begin{aligned}
\Gamma_{\mathsf{e}(x)\twoheadrightarrow\mathsf{true}\,\&\,\mathsf{o}(x)\twoheadrightarrow\mathsf{true}} &\to \textsc{Rec}(\Gamma_{\mathsf{e}(x')\twoheadrightarrow\mathsf{true}}, \{x \mapsto x'\}) \,\&\, \textsc{Rec}(\Gamma_{\mathsf{o}(x'')\twoheadrightarrow\mathsf{true}}, \{x \mapsto x''\})\\
\Gamma_{\mathsf{e}(x')\twoheadrightarrow\mathsf{true}} &\to id \,\&\{x' \mapsto \mathsf{0}\} \mid \big(\textsc{Rec}(\Gamma_{\mathsf{o}(x'')\twoheadrightarrow\mathsf{true}}, \{x_1 \mapsto x''\}) \,\&\, id\big) \,\&\{x' \mapsto \mathsf{s}(x_1)\}\\
&\quad \mid \big(\textsc{Rec}(\Gamma_{\mathsf{e}(x')\twoheadrightarrow\mathsf{true}}, \{x_2 \mapsto x'\}) \,\&\, \varnothing\big) \,\&\{x' \mapsto \mathsf{s}(x_2)\}\\
\Gamma_{\mathsf{o}(x'')\twoheadrightarrow\mathsf{true}} &\to \varnothing \,\&\{x'' \mapsto \mathsf{0}\} \mid \big(\textsc{Rec}(\Gamma_{\mathsf{e}(x')\twoheadrightarrow\mathsf{true}}, \{x_3 \mapsto x'\}) \,\&\, id\big) \,\&\{x'' \mapsto \mathsf{s}(x_3)\}\\
&\quad \mid \big(\textsc{Rec}(\Gamma_{\mathsf{o}(x'')\twoheadrightarrow\mathsf{true}}, \{x_4 \mapsto x''\}) \,\&\, \varnothing\big) \,\&\{x'' \mapsto \mathsf{s}(x_4)\}
\end{aligned}$$

These rules can be simplified to $\Gamma_{\mathsf{e}(x)\twoheadrightarrow\mathsf{true}\,\&\,\mathsf{o}(x)\twoheadrightarrow\mathsf{true}} \to \varnothing$, and the critical pair is infeasible.

To prove infeasibility of a condition c, the tool first prove 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.

References

[1] K. Gmeiner, B. Gramlich, and F. Schernhammer. On soundness conditions for unraveling deterministic conditional rewrite systems. In *Proc. RTA 2012*, vol. 15 of *LIPIcs*, pp. 193–208, 2012.

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

[3] Y. Maeda, N. Nishida, M. Sakai, and T. Kobayashi. Extending narrowing trees to basic narrowing in term rewriting. IEICE Tech. Rep. SS2018-39, Vol. 118, No. 385, pp. 73–78, 2019, in Japanese.

[4] N. Nishida, T. Kuroda, and K. Gmeiner. CO3 (Version 1.3). In *Proc. IWC 2016*, p. 74, 2016.

[5] N. Nishida and Y. Maeda. Narrowing trees for syntactically deterministic conditional term rewriting systems. In *Proc. FSCD 2018*, vol. 108 of *LIPIcs*, pp. 26:1–26:20, 2018.

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

[1] http://www.trs.css.i.nagoya-u.ac.jp/co3/