CRaris (Version 1.1)

Naoki Nishida and Misaki Kojima

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

CRaris, a **CR** checker for LCTRSs in **ARI s**tyle,[1] is a tool to prove confluence of *logically constrained term rewrite systems* (LCTRSs, for short) [5] written in ARI format [1].[2] The tool is based on Crisys2, **c**onstrained **r**ewriting **i**nduction **sys**tem (version **2**),[3] and receives LCTRSs written in ARI format only to prove confluence, while Crisys2 has many functions to e.g., solve *all-path reachability problems* [3]. To prove confluence of LCTRSs, the tool uses the following criteria:

- weak orthogonality [5], and
- termination and joinability of critical pairs [8].

To prove termination, the tool uses the DP framework for LCTRSs [4] without any interpretation method, together with a criterion for LCTRSs with bitvector arithmetics [6].

The *critical pairs* of two constrained rewrite rules $\rho_1 : \ell_1 \to r_1\ [\varphi_1]$ and $\rho_2 : \ell_2 \to r_2\ [\varphi_2]$ with distinct variables (i.e., $\mathcal{V}ar(\ell_1, r_1, \varphi_1) \cap \mathcal{V}ar(\ell_2, r_2, \varphi_2) = \emptyset$) are all tuples $\langle s, t, \phi\rangle$ such that a non-variable subterm $\ell_1|_p$ of ℓ_1 at a position p is unifiable with ℓ_2, "$p \neq \varepsilon$, $\rho_1 \neq \rho_2$ up to variable renaming, or $\mathcal{V}ar(r_1) \subseteq \mathcal{V}ar(\ell_1)$", the most general unifier γ of $\ell_1|_p$ and ℓ_2 respects variables of both ρ_1 and ρ_2, i.e., $\gamma(x)$ is either a value or a variable for all variables x in $\mathcal{V}ar(\varphi_1, \varphi_2) \cup (\mathcal{V}ar(r_1, r_2) \setminus \mathcal{V}ar(\ell_1, \ell_2))$, $(\varphi_1 \wedge \varphi_2)\gamma$ is satisfiable, $s = r_1\gamma$, $t = (\ell_1[r_2]_p)\gamma$, and $\phi = (\varphi_1 \wedge \varphi_2)\gamma$. The set of critical pairs of an LCTRS $\mathcal{R}$ is denoted by $CP(\mathcal{R})$, which includes all critical pairs of two rules in $\mathcal{R} \cup \mathcal{R}_{calc}$. A critical pair $\langle s, t, \phi\rangle$ is called *trivial* if $s\ [\phi] \sim t\ [\phi]$. An LCTRS $\mathcal{R}$ is called *weakly orthogonal* if $\mathcal{R}$ is left-linear and all critical pairs of $\mathcal{R}$ are trivial.

Theorem 1 ([5])**.** *A weakly orthogonal LCTRS is confluent.*

A critical pair $\langle s, t, \phi\rangle$ is called *joinable* if $(\langle s, t\rangle\ [\phi]) \to^*_{\mathcal{R}} (\langle s', t'\rangle\ [\phi'])$ and $s'\ [\phi'] \sim t'\ [\phi']$.

Theorem 2 ([8])**.** *A terminating LCTRS is confluent if all its critical pairs are joinable.*

The previous version [7] uses syntactic equivalence of terms as a sufficient condition for a critical pair $\langle s, t, \phi\rangle$ being trivial.

Proposition 3 ([7])**.** *A critical pair $\langle s, s, \phi\rangle$ is trivial and thus joinable.*

This version uses the idea of EQ-DELETION of constrained rewriting induction [2].

Proposition 4. *A critical pair $\langle s, t, \phi\rangle$ is trivial if there exist positions $p_1, \ldots, p_n$ of s such that $p_1, \ldots, p_n$ are positions of t, $t = s[t_1, \ldots, t_n]_{p_1,\ldots,p_n}$, $s|_{p_1}, t_1 \ldots, s|_{p_n}, t_n$ are theory terms, $\mathcal{V}ar(s|_{p_1}, \ldots, s|_{p_n}, t_1, \ldots, t_n) \subseteq \mathcal{V}ar(\phi)$, and $\phi \wedge \neg(\bigwedge_{i=1}^{n}(s|_{p_i} = t_i))$ is unsatisfiable.*

In addition, this version uses a very simple variant of the disproof criterion—*an LCTRS is not confluent if there exists a constrained critical pair that rewrites to a non-trivial constrained equation in normal form*—in [9, Lemma 1].

Proposition 5. *An LCTRS $\mathcal{R}$ is not confluent if there exists a critical pair $\langle s, t, \phi\rangle$ of $\mathcal{R}$ such that s, t are variables in $\mathcal{V}ar(\phi)$ and $\phi \wedge \neg(\bigwedge_{i=1}^{n}(s|_{p_i} = t_i))$ is satisfiable.*

[1] http://www.trs.css.i.nagoya-u.ac.jp/craris/
[2] https://project-coco.uibk.ac.at/ARI/lctrs.php
[3] https://www.trs.cm.is.nagoya-u.ac.jp/crisys/

References

[1] T. Aoto, N. Hirokawa, D. Kim, M. Kojima, A. Middeldorp, F. Mitterwallner, N. Nishida, T. Saito, J. Schöpf, K. Shintani, R. Thiemann, and A. Yamada. A new format for rewrite systems. In C. Chenavier and S. Winkler, editors, *Proceedings of the 12th International Workshop on Confluence*, pages 32–37, 2023.

[2] C. Fuhs, C. Kop, and N. Nishida. Verifying procedural programs via constrained rewriting induction. *ACM Transactions on Computational Logic*, 18(2):14:1–14:50, 2017.

[3] M. Kojima and N. Nishida. Reducing non-occurrence of specified runtime errors to all-path reachability problems of constrained rewriting. *Journal of Logical and Algebraic Methods in Programming*, 135:1–19, 2023.

[4] C. Kop. Termination of LCTRSs. In *Proceedings of the 13th International Workshop on Termination*, pages 1–5, 2013.

[5] C. Kop and N. Nishida. Term rewriting with logical constraints. In P. Fontaine, C. Ringeissen, and R. A. Schmidt, editors, *Proceedings of the 9th International Symposium on Frontiers of Combining Systems*, volume 8152 of *Lecture Notes in Artificial Intelligence*, pages 343–358. Springer, 2013.

[6] A. Matsumi, N. Nishida, M. Kojima, and D. Shin. On singleton self-loop removal for termination of LCTRSs with bit-vector arithmetic. In A. Yamada, editor, *Proceedings of the 19th International Workshop on Termination*, pages 1–6, 2023.

[7] N. Nishida and M. Kojima. CRaris: CR checker for LCTRSs in ARI Style. In C. Chenavier and N. Nishida, editors, *Proceedings of the 13th International Workshop on Confluence*, pages 83–84, 2024.

[8] J. Schöpf and A. Middeldorp. Confluence criteria for logically constrained rewrite systems. In B. Pientka and C. Tinelli, editors, *Proceedings of the 29th International Conference on Automated Deduction*, volume 14132 of *Lecture Notes in Computer Science*, pages 474–490. Springer, 2023.

[9] J. Schöpf and A. Middeldorp. Automated analysis of logically constrained rewrite systems using crest. In A. Gurfinkel and M. Heule, editors, *Proceedings of the 31st International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, volume 15696 of *Lecture Notes in Computer Science*, pages 124–144. Springer, 2025.