CoCo 2016 Participant: CeTA 2.28*

Julian Nagele, Christian Sternagel, and Thomas Sternagel

Department of Computer Science, University of Innsbruck, Austria

Automatic provers have become popular in many areas like theorem proving, SMT, etc. Since such provers are complex pieces of software, they might contain errors that lead to wrong answers, i.e., incorrect proofs. Therefore, certification of the generated proofs is of major importance.

The tool CeTA [7] is a certifier that can be used to certify confluence and non-confluence proofs of term rewrite systems (TRSs) and conditional term rewrite systems (CTRSs). Its soundness is proven as part of IsaFoR, the *Isabelle Formalization of Rewriting*. The following techniques are currently supported in CeTA—for further details we refer to the certification problem format (CPF) and to the sources of IsaFoR and CeTA (http://cl-informatik.uibk.ac.at/software/ceta/).

Term rewrite systems. Since CeTA was originally conceived for termination analysis, our first method is Newman's lemma in combination with the critical pair theorem. For possibly non-terminating TRSs, CeTA can ensure that weakly orthogonal, strongly closed, and almost parallel closed TRSs are confluent [4], as well as check applications of the rule labeling heuristic [5] and addition and removal of redundant rules [3]. To certify non-confluence one can provide a divergence and a certificate for non-joinability. Here CeTA supports: distinct normal forms, *tcap*, usable rules, discrimination pairs, argument filters and interpretations [1], and reachability analysis using tree automata techniques [2].

Conditional term rewrite systems. Since last year CeTA also supports confluence criteria for conditional rewriting. CeTA can certify that almost orthogonal, extended properly oriented, right-stable 3-CTRSs are confluent, including support for infeasible critical pairs, where the supported justification is a certificate for non-reachability using either *tcap* or tree automata [6]. The second supported technique for CTRSs is unraveling [8], transforming the system into a TRS where then the aforementioned techniques can be certified.

References

[1] T. Aoto. Disproving confluence of term rewriting systems by interpretation and ordering. In *FroCoS*, volume 8152 of *LNCS*, pages 311–326, 2013.

[2] B. Felgenhauer and R. Thiemann. Reachability, confluence, and termination analysis with state-compatible automata. *I&C*, 2016. Available Online.

[3] J. Nagele, B. Felgenhauer, and A. Middeldorp. Improving automatic confluence analysis of rewrite systems by redundant rules. In *RTA*, volume 36 of *LIPIcs*, pages 257–268, 2015.

[4] J. Nagele and A. Middeldorp. Certification of classical confluence results for left-linear term rewrite systems. In *ITP*, volume 9807 of *LNCS*, pages 290–306, 2016.

[5] J. Nagele and H. Zankl. Certified rule labeling. In *RTA*, volume 36 of *LIPIcs*, pages 269–284, 2015.

[6] C. Sternagel and T. Sternagel. Certifying confluence of almost orthogonal CTRSs via exact tree automata completion. In *FSCD*, volume 52 of *LIPIcs*, pages 29:1–29:16, 2016.

[7] R. Thiemann and C. Sternagel. Certification of termination proofs using CeTA. In *TPHOLs*, volume 5674 of *LNCS*, pages 452–468, 2009.

[8] S. Winkler and R. Thiemann. Formalizing soundness and completeness of unravelings. In *FroCoS*, volume 9322 of *LNCS (LNAI)*, 2015.

*Supported by Austrian Science Fund (FWF), projects P27502, and P27528.