

CoScart: Confluence Prover in Scala

Karl Gmeiner¹

UAS Technikum Wien, Vienna, Austria
`gmeiner@technikum-wien.at`

1 Overview

CoScart is a tool to prove confluence of first-order term rewrite systems and deterministic conditional term rewrite systems automatically. It originates from the project *ScaRT* that consists of multiple classes for term rewriting. The main purpose of this project is to experimentally compare transformations of conditional term rewrite systems and implement program transformations for in particular functional programming languages.

Scart itself is a reimplementaion of this project that was originally written in Java (under the working title “KaRT”). A first version was used to conduct the experiments in [2]. To speed up and simplify development, in particular with focus on implementing CoScart, the whole project was ported to Scala recently, a functional, object-oriented programming language.

2 Technical Details

Scart itself only supported higher-order rewrite systems originally. Support for first order terms was only added to simplify the development of CoScart. Therefore, a future support of HORSs is the highest priority for the future development.

Scart contains a rewrite engine that stores and rewrites DAGs of terms in a list. This turns out to be a very efficient way concerning time and memory.

In order to use the Knuth-Bendix method to prove confluence, Scart contains an automatic termination prover for first-order TRSs that uses the dependency pairs method in combination with argument filterings with the *some more*-heuristics of [1].

Since CoScart is currently a one-man project, there are no sophisticated user interfaces yet. Scart is available at <https://github.com/searles/RewriteTool/>.

2.1 Implemented Methods

CoScart proves confluence of (deterministic conditional) TRSs using the following methods: Transformation of [3] from DCTRSs into TRSs, modularity of confluence, Knuth-Bendix, and development-closed critical pairs of left-linear TRSs.

References

- [1] N. Hirokawa and A. Middeldorp. Automating the Dependency Pair Method. In *Proc. CADE 2003*, LNAI vol. 2741, pp. 32–46, Springer-Verlag, 2003.
- [2] K. Gmeiner and B. Gramlich. Transformations of Conditional Rewrite Systems Revisited. In *Proc. WADT 2008*, LNCS vol. 5486, pp. 166-186, Springer-Verlag, 2009.
- [3] K. Gmeiner and N. Nishida. Notes on Structure-Preserving Transformations of Conditional Term Rewrite Systems. In *Proc. WPTTE 2014*, OASiCs vol. 40, pp. 3–14, 2014.