Modularity Results for Termination Proofs using Dependency Pairs

Jürgen Giesl
Department of Computer Science, TU Darmstadt
Alexanderstr. 10, 64283 Darmstadt, Germany
E-mail: giesl@informatik.tu-darmstadt.de

Thomas Arts
Computer Science Laboratory, Ericsson Telecom AB
126 25 Stockholm, Sweden
E-mail: thomas@cslab.ericsson.se

Enno Ohlebusch
Faculty of Technology, University of Bielefeld
P.O. Box 10 01 31, 33501 Bielefeld, Germany
E-mail: enno@TechFak.Uni-Bielefeld.DE

Most classical techniques for automated termination proofs of term rewriting systems (TRSs) are based on simplification orderings. In practice, however, many TRSs are non-simply terminating, i.e., their termination cannot be verified by any simplification ordering. For that reason, we developed the dependency pair approach [AG97a, AG97b, AG99] which allows automated termination proofs for many TRSs where such proofs were not possible before.

In the talk we present a refinement of the dependency pair framework which further extends the class of term rewriting systems for which termination can be proved automatically. By means of this refinement, one can now prove termination in a modular way. To be more precise, in contrast to other methods, one may use several different orderings in one termination proof, cf. [AG98].

Subsequently, we present new modularity results based on dependency pairs. It it is well known that simple termination is modular for certain kinds of combinations of TRSs. This result is of practical relevance, because it ensures modularity whenever the previous standard techniques for automated termination proofs succeed. But by the development of dependency pairs, the class of TRSs where termination is provable automatically has been increased considerably. The reason is that by using (quasi-)simplification orderings in combination with dependency pairs, it is now possible to prove termination of non-simply terminating systems automatically. Thus, it is natural to ask whether modularity of simple termination can be extended to those systems which can be handled by this new technique. In our talk we show that for disjoint unions this is indeed the case, cf. [GO98]. Moreover, under certain additional conditions, this result also holds for constructor-sharing and composable systems.

References

[AG97a] T. Arts and J. Giesl. Automatically proving termination where simplification orderings fail. In M. Dauchet, editor, *Proceedings of the 7th International Joint Conference on the Theory and Practice of Software Development, TAPSOFT '97*, volume 1214 of *Lecture Notes in Computer Science*, pages 261–272, Lille, France, April 1997. Springer Verlag, Berlin.

[AG97b] T. Arts and J. Giesl. Proving innermost normalisation automatically. In H. Comon, editor, *Proceedings of the 8th International Conference on Rewriting Techniques and Applications, RTA-97*, volume 1232 of *Lecture Notes in Computer Science*, pages 157–171, Sitges, Spain, June 1997. Springer Verlag, Berlin.

[AG98] T. Arts and J. Giesl. Modularity of termination using dependency pairs. In T. Nipkow, editor, *Proceedings of the 9th International Conference on Rewriting Techniques and Applications, RTA-98*, volume 1379 of *Lecture Notes in Computer Science*, pages 226–240, Tsukuba, Japan, March/April 1998. Springer Verlag, Berlin.

[AG99] T. Arts and J. Giesl. Termination of term rewriting using dependency pairs. *Theoretical Computer Science*, 1999. To appear.

[GO98] J. Giesl and E. Ohlebusch. Pushing the frontiers of combining rewrite systems farther outwards. In *Proceedings of the Second International Workshop on Frontiers of Combining Systems, FroCoS '98*, Applied Logic Series, Amsterdam, The Netherlands, October 1998. Kluwer Academic Publishers, Amsterdam.