Automatic Termination Proofs of Logic Programs via Rewrite Systems

Enno Ohlebusch

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

In the last decade, several methods have been proposed to check termination of logic programs (semi-)automatically. In [GW93], Ganzinger and Waldmann described a transformation of well-moded logic programs into deterministic conditional term rewriting systems–systems with a restricted use of extra-variables. If the transformed system $\mathcal{R}(P)$ is quasi-reductive (a property which ensures effective termination), then the logic program P is terminating for all well-moded queries but not vice versa (so the method is sound but not complete). Quasi-reductivity is in general undecidable but sufficient conditions for quasi-reductivity are known; see [Gan91, ALS94].

Inspired by the transformational approach described above, Arts and Zantema [AZ95, AZ96] proposed to transform the logic program into an unconditional term rewriting system. A similar approach appeared in [AM93]. In contrast to the former techniques, the latter does not require any prior information about modes of predicates because these are computed during the transformation according to a given query.

In Arts and Zantema's method, innermost termination of the term rewriting system ensures termination of the logic program. The advantage is that well-known powerful methods like the dependency pair approach [AG97] can now be used to prove termination of logic programs automatically. Consequently, it is remarked in [AZ95] that the suggested method "is applicable to a wider class of logic programs" and hence it is "stronger than the other results". It will be shown, however, that the technique of Ganzinger and Waldmann is equally powerful. This is done by first translating the deterministic conditional term rewriting system $\mathcal{R}(P)$ into an unconditional term rewriting system $U(\mathcal{R}(P))$. Then one can show that quasi-reductivity of $\mathcal{R}(P)$ coincides with innermost termination of $U(\mathcal{R}(P))$.

References

[AG97] T. Arts and J. Giesl. Automatically Proving Termination where Simplification Orderings Fail. In *Proceedings of the 22nd International Colloquium on Trees in Algebra and Programming*, pages 261–273. LNCS **1214**, 1997.

[ALS94] J. Avenhaus and C. Loría-Sáenz. On Conditional Rewrite Systems with Extra Variables and Deterministic Logic Programs. In *Proceedings of the 5th International Conference on Logic Programming and Automated Reasoning*, pages 215–229. LNAI **822**, 1994.

[AM93] G. Aguzzi and U. Modigliani. Proving Termination of Logic Programs by Transforming them into Equivalent Term Rewriting Systems. In *Proceedings of the 13th Conference on the Foundations of Software Technology and Theoretical Computer Science*, pages 114–124. LNCS **761**, 1993.

[AZ95] T. Arts and H. Zantema. Termination of Logic Programs via Labelled Term Rewrite Systems. In *Proceedings of Computing Science in the Netherlands*, pages 22–34, 1995.

[AZ96] T. Arts and H. Zantema. Termination of Logic Programs using Semantic Unification. In *Proceedings of the Fifth International Workshop on Logic Program Synthesis and Transformation*, pages 219–233. LNCS **1048**, 1996.

[Gan91] H. Ganzinger. Order-Sorted Completion: The Many-Sorted Way. *Theoretical Computer Science* **89**, pages 3–32, 1991.

[GW93] H. Ganzinger and U. Waldmann. Termination Proofs of Well-Moded Logic Programs via Conditional Rewrite Systems. In *Proceedings of the 3rd International Workshop on Conditional Term Rewriting Systems 1992*, pages 113–127. LNCS **656**, 1993.