

Fast and Accurate Strong Termination Analysis with an Application to Partial Evaluation

Michael Leuschel¹, Salvador Tamarit², and Germán Vidal²

¹ Institut für Informatik, Universität Düsseldorf, D-40225, Düsseldorf, Germany
`leuschel@cs.uni-duesseldorf.de`

² DSIC, Technical University of Valencia, E-46022, Valencia, Spain
`{stamarit,gvidal}@dsic.upv.es`

Analysing the termination of logic programs is a challenging problem that has attracted a lot of interest (see, e.g., [2, 3, 7, 8] and references therein). However, *strong* termination analysis (i.e., termination for any selection rule) has received little attention, a notable exception being the work by Bezem [1].

A logic program strongly terminates if it terminates for any selection rule. Clearly, considering a particular selection rule—like Prolog’s leftmost selection rule—allows one to prove more goals terminating. In contrast, a strong termination analysis gives valuable information for those applications in which the selection rule cannot be fixed in advance (e.g., partial evaluation, dynamic selection rules, parallel execution).

In this paper, we introduce a fast and accurate size-change analysis [4] that can be used to infer conditions for both strong termination and quasi-termination of logic programs. In the paper we also provide several ways to increase the accuracy of the analysis without sacrificing scalability. In the experimental evaluation, we show that the new algorithm is up to three orders of magnitude faster than the previous implementation, meaning that we can efficiently deal with realistic programs exceeding 25,000 lines of Prolog.

One major application of this algorithm is for offline partial evaluation of large programs. In the experimental evaluation we show that, with our new algorithm, we can now deal with realistic interpreters, such as the interpreter for the full B specification language from [6]. Together with the selective use of hints [5], we have obtained both a scalable and an effective partial evaluation procedure.

References

1. M. Bezem. Strong Termination of Logic Programs. *Journal of Logic Programming*, 15(1&2):79–97, 1993.
2. M. Codish and C. Taboch. A semantic basis for the termination analysis of logic programs. *Journal of Logic Programming*, 41(1):103–123, 1999.
3. Danny De Schreye and Stefaan Decorte. Termination of logic programs: The never ending story. *The Journal of Logic Programming*, 19 & 20:199–260, May 1994.
4. C.S. Lee, N.D. Jones, and A.M. Ben-Amram. The Size-Change Principle for Program Termination. *SIGPLAN Notices (Proc. of POPL’01)*, 28:81–92, 2001.
5. M. Leuschel and G. Vidal. Fast Offline Partial Evaluation of Large Logic Programs. In *Proc. of LOPSTR’08*, pages 119–134. Springer LNCS 5438, 2009.
6. M. Leuschel and M. Butler. ProB: A model checker for B. *Proceedings FME 2003*, LNCS 2805, pages 855–874. Springer-Verlag, 2003.
7. Naomi Lindenstrauss, Yehoshua Sagiv, and Alexander Serebrenik. Proving Termination for Logic Programs by the Query-Mapping Pairs Approach. In *Program Development in Computational Logic*, pages 453–498, 2004.
8. S. Verbaeten, K. Sagonas, and D. De Schreye. Termination Proofs for Logic Programs with Tabling. *ACM Transactions on Computational Logic*, 2(1):57–92, 2001.