On Solving Termination Constraints

Evelyne Contejean Claude Marché Ana-Paula Tomás
Xavier Urbain

September 1, 1999

We are interested in automatically proving termination of rewrite systems, and to design a procedure to do so in modular way, we introduce *termination constraints*. A solution of such a constraint is a term ordering.

The termination problem of a rewrite system can be translated into termination constraints by various termination criteria, such as the well-known Manna & Ness criterion (each lhs of rules has to be greater than the rhs, and the ordering has to be stable and monotonic), or the dependency pair criteria proposed by Arts and Giesl [1, 2].

Atomic termination constraints are either of the form $s > t$ or $s \geq t$ for some terms s and t, or properties that $\geq$ should satisfy: weak or strict monotonicity with respect to some function symbols, compatibility with associativity and/or commutativity of some symbols, etc.

Because of undecidability problems, solving termination constraints in practice is by looking for a term ordering only in a given class of orderings. In a second part of this work, we show a method for solving these termination constraints in the domain of orderings defined by polynomial interpretations.

When looking for an ordering in this class, termination constraints must first be translated into Diophantine constraints. We improve the methods proposed by Steinbach [5] and Giesl [4] in two ways: first we improve the translation by using new criterion for positivity of a polynomial, and second for solving the resulting Diophantine constraints, we translate them into finite domain constraints, for which quite efficient solving techniques are known. In this last translation arises a problem very similar to the so-called *addition chains* algorithmic problem, and we propose a partial solution to this problem.

The figure 1 summarizes the sequence of translations made in the termination procedure. This termination procedure has been implemented in the system CiME [3][1] and we show interesting experimental results.

References

[1] T. Arts and J. Giesl. Automatically proving termination where simplification orderings fail. In Michel Bidoit and Max Dauchet, editors, *Theory*

[1] Available at `http://www.lri.fr/~demons/cime-2.0.html`

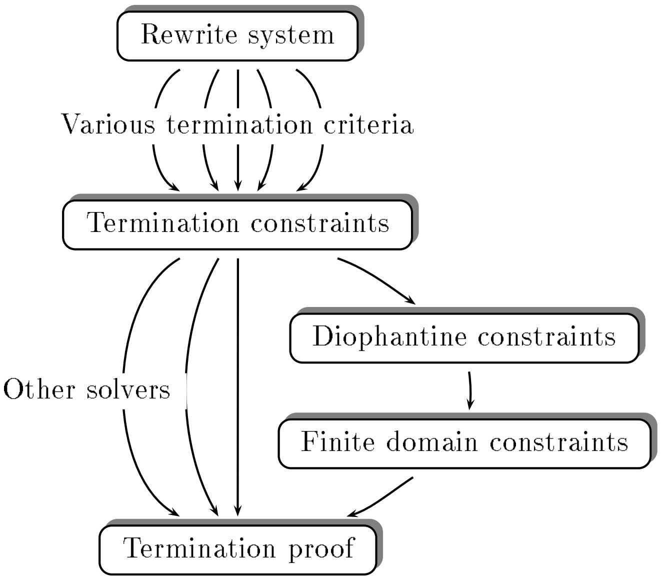

Figure 1: General view of the termination procedure

and Practice of Software Development, volume 1214 of *Lecture Notes in Computer Science*, Lille, France, April 1997. Springer-Verlag.

[2] Thomas Arts and Jürgen Giesl. Modularity of termination using dependency pairs. In Tobias Nipkow, editor, *9th International Conference on Rewriting Techniques and Applications*, volume 1379 of *Lecture Notes in Computer Science*, pages 226–240, Tsukuba, Japan, April 1998. Springer-Verlag.

[3] Evelyne Contejean and Claude Marché. CiME: Completion Modulo E. In Harald Ganzinger, editor, *7th International Conference on Rewriting Techniques and Applications*, volume 1103 of *Lecture Notes in Computer Science*, pages 416–419, New Brunswick, NJ, USA, July 1996. Springer-Verlag. System Description available at `http://www.lri.fr/~demons/cime.html`.

[4] Jürgen Giesl. Generating polynomial orderings for termination proofs. In Jieh Hsiang, editor, *6th International Conference on Rewriting Techniques and Applications*, volume 914 of *Lecture Notes in Computer Science*, Kaiserslautern, Germany, April 1995. Springer-Verlag.

[5] Joachim Steinbach. Generating polynomial orderings. *Information Processing Letters*, 49:85–93, 1994.