Structural Complexity of Term Schematizations

(Abstract)

Gernot Salzer
Technische Universität Wien
e-mail: salzer@logic.tuwien.ac.at

Within the last years a zoo of different species of schematizations have been proposed in literature, like ω-terms [1], I-terms [2], R-terms [6] and primal grammars [4]. The aim of these schematizations is to finitely describe infinite sequences of structurally similar terms arising in clausal theorem proving and during completion in term rewriting, with the hope to avoid infinite loops and divergence.

To be really applicable term schematizations have to be finitely unifiable. Unifying two schematizations t_1 and t_2 means to check every first order term schematized by t_1 against every one schematized by t_2 for unifiability. In general there is an infinite number of unifiable pairs of terms and therefore an infinite number of first order unifiers. There are two things one could expect from a unification algorithm for term schematizations. First, it should be able to decide whether t_1 and t_2 are unifiable at all, i.e., whether there is any pair of terms schematized by t_1 and t_2 that can be first order unified. Secondly, the unification algorithm should compute some finite representation of the infinitely many first order unifiers, using again term schematizations. All formalisms listed above are well-behaved in this sense: their unification is decidable and finitary (in the language of schematizations).

Two related problems have not been considered so far, namely

- the word problem for term schematizations, and
- the complexity of the unification problem.

The word problem is interesting for several reasons. First, being able to determine whether two objects are identical on the semantic level though differing syntactically is a minimal requirement for eliminating redundancies in any deduction process. Secondly, all unification algorithms presented so far do not recognize semantically equivalent schematizations. As a result they produce more unifiers than necessary. Keeping the schematizations normalized during unification would lead to smaller sets of unifiers. Finally, knowing the complexity of the word problem is useful in obtaining upper bounds for the complexity of unification.

One reason for not investigating the complexity of unification are Diophantine equations: term schematizations are able to encode rather directly linear Diophantine equations over integers, and therefore inherit the complexity of these equations. In particular, the unification of schematizations requires as a sub-task the solution of linear Diophantine equations. For many purposes, however, the structure of the occurring Diophantine equations is rather simple. Therefore it seems to be a natural question to ask for the complexity of unification when *disregarding the Diophantine part.*

Results

Generalizing the algorithm for R-strings [5] we obtain an algorithm solving the word problem for R-terms. The word problem becomes polynomially solvable when representing term schematizations as directed acyclic graphs.

A lower bound for the complexity of unification is obtained by coding the 1-in-3-SAT problem [3] as unification problem for term schematizations. The coding can be done in a way such that every numeric variable occurs at most once, i.e., the unification problem for term schematizations is NP-hard even when no Diophantine equations have to be solved. Analyzing the unification algorithm for R-terms it can be shown that there is always a unifier of polynomial length (represented as DAG) provided the given unification problem is solvable. Taking into account that the word problem is polynomially solvable, we obtain that the unification problem is in NP, i.e.: *the unification problem for R-terms is NP-complete when assigning unit cost to the solution of linear Diophantine equations.*

References

[1] H. Chen and J. Hsiang. Logic programming with recurrence domains. In J. Leach Albert, B. Monien, and M. Rodríguez Artalejo, editors, *Proc. 18th ICALP Conf.*, LNCS 510, pages 20–34. Springer, July 1991.

[2] H. Comon. Completion of rewrite systems with membership constraints. In W. Kuich, editor, *Proc. 19th ICALP Conf*, LNCS 623, pages 392–403. Springer, July 1992.

[3] M. R. Garey and D. S. Johnson. *Computers and Intractability.* W. H. Freeman and Company, New York, 1979.

[4] M. Hermann and R. Galbavý. Unification of infinite sets of terms schematized by primal grammars. *Theoretical Computer Science*, 1996. To appear.

[5] G. Salzer. Deductive generalization and meta-reasoning, or how to formalize Genesis. In *Proc. 7. Österreichische Artificial Intelligence Tagung*, Informatik-Berichte 287, pages 103–115. Springer, 1991.

[6] G. Salzer. The unification of infinite sets of terms and its applications. In A. Voronkov, editor, *Logic Programming and Automated Reasoning (LPAR'92)*, LNCS 624 (LNAI), pages 409–420. Springer, 1992.