Term Schematizations and Their Applications

Proposal for a One-Day-Workshop

M. Hermann*　　G. Salzer†

November 23, 1995

A common phenomenon throughout all areas dealing with first order terms—like clausal theorem proving, logic programming or equational reasoning, to name but a few—are *infinite sequences of structurally similar terms.* These sequences arise from self-resolving clauses acting as loops (leading to infinite branches in search space), during completion of term rewrite systems (leading to divergence), or may be part of an infinite Herbrand model. One characteristic of (applied) computer science is its emphasis on finite structures because this is what real computers can deal with. Consequently, various formalisms and methods have been proposed to capture infinite sequences by finite means, among them *schematizations of terms.*

A schematization of an infinite set of terms is a finite expression together with a mechanism for effectively generating the elements of the set. In addition, schematizations should act as representatives for the corresponding infinite sets with regard to some basic operations. In particular, it should be possible to compute the intersection as well as the unifying intersection of infinite term sets by manipulating their finite schematizations in place of the sets. Typically, a unification algorithm for term schematizations computes a finite set of unifiers—again expressed in the language and formalism of the schematizations—representing an infinite number of ordinary unifiers. From this very general point of view, a first order term can be understood as a finite schematization of all its ground instances; unification of two terms then means the intersection (which is equivalent to the unifying intersection in this simple case) of the corresponding sets of ground instances.

Within the last years several kinds of schematizations have been proposed. The different approaches range from meta-rules [12], term schemes [8], membership constraints [4, 6], stratified second-order terms [25] to congruence grammars [13] and recurrence terms [1, 2, 5, 9–11, 18–20].

*CRIN (CNRS) and INRIA-Lorraine, Campus Scientifique, BP 239, 54506 Vandœuvre-lès-Nancy, France; Email: `Miki.Hermann@loria.fr`

†Technische Universität Wien, Karlsplatz 13/E185-2, A-1040 Vienna, Austria; Email: `salzer@logic.tuwien.ac.at`

The applications of term schematizations are not yet as well explored as the schematizations themselves. Potentially, every field based on first order terms is a candidate for the use of schematizations.

Clausal Theorem Proving. As has been shown in [18, 20], schematizations can be constructed automatically and may lead to considerably shorter proofs in resolution-style theorem proving.

Logic Programming. Similarly, logic programming might profit from the use of schematizations by speeding up long computations or avoiding infinite loops [1]. In particular, schematizations could enhance loop dedection during the partial evaluation of logic programs [16, 17].

Database Theory. Schematizations can be used as finite and safe extensions of Datalog [3, 15].

Term Rewriting. In [10] quite natural divergent term rewrite systems are presented, where the introduction of schematizations leads to finite and complete rewrite systems.

Unification Theory. Another area of application is unification modulo a binary clause, where term schematizations can be used to decide certain classes of unification problems [21–23]. Furthermore, the work of Contejean [7] and the recent positive solution of unifiability with distributivity [24] indicate that schematizations might be helpful in this area.

Higher-Order Unification. Schematizations can be used to discover decidable subclasses of higher-order unification [25].

Finite Model Building. Recently, term schematizations are also being used in model building to obtain finite representations of infinite models [14].

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. Chen, J. Hsiang, and H.-C. Kong. On finite representations of infinite sequences of terms. In S. Kaplan and M. Okada, editors, *Proc. 2nd Int. Workshop on Conditional and Typed Rewriting Systems*, LNCS 516, pages 100–114. Springer, June 1990.

[3] J. Chomicki and T. Imieliński. Finite representation of infinite query answers. *ACM Transactions on Database Systems*, 18(2):181–223, 1993.

[4] 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.

[5] H. Comon. On unification of terms with integer exponents. *Mathematical Systems Theory*, 28(1):67–88, 1995.

[6] H. Comon and C. Delor. Equational formulae with membership constraints. *Information and Computation*, 112(2):167–216, 1994.

[7] E. Contejean. Solving $*$-problems modulo distributivity by a reduction to AC1-unification. *Journal of Symbolic Computation*, 16(5):493–521, 1993.

[8] B. Gramlich. Unification of term schemes - theory and applications. SEKI Report SR-88-18, Universität Kaiserslautern, Germany, 1988.

[9] M. Hermann. On the relation between primitive recursion, schematization, and divergence. In H. Kirchner and G. Levi, editors, *Proc. 3rd Conf. on Algebraic and Logic Programming*, LNCS 632, pages 115–127. Springer, September 1992.

[10] M. Hermann. *Divergence des Systèmes de Réécriture et Schématisation des Ensembles Infinis de Termes.* Habilitation, Université de Nancy I and CRIN-CNRS Inria Lorraine, Nancy (France), 1994.

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

[12] H. Kirchner. Schematization of infinite sets of rewrite rules generated by divergent completion process. *Theoretical Computer Science*, 67(2-3):303–332, October 1989.

[13] D. McAllester. Grammar rewriting. In D. Kapur, editor, *Proc. 11th Int. Conf. on Automated Deduction*, LNCS 607 (LNAI), pages 124–138. Springer, June 1992.

[14] N. Peltier. Model building by constraint solving with terms with integer exponents. Unpublished manuscript, 1995.

[15] Y. Sagiv and M. Y. Vardi. Safety of Datalog queries over infinite databases. In *Proc. 8th Symp. on Principles of Database Systems*, pages 160–171, 1989.

[16] D. Sahlin. The Mixtus approach to automatic partial evaluation of full Prolog. In S. Debray and M. Hermenegildo, editors, *Logic Programming: Proc. 1990 North American Conf.*, pages 377–398. MIT Press, 1990.

[17] D. Sahlin. *An Automatic Partial Evaluator for Full Prolog.* Dissertation, Kungliga Tekniska Högskolan, Stockholm, Sweden, 1991.

[18] 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.

[19] G. Salzer. *Unification of Meta-Terms.* PhD thesis, Technische Universität Wien, Austria, 1991.

[20] 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.

[21] G. Salzer. On the relationship between cycle unification and the unification of infinite sets of terms. In F. Baader and W. Snyder, editors, *7th Workshop on Unification (UNIF'93)*, Boston, Mass., USA, June 13–14, 1993.

[22] G. Salzer. Primal grammars and unification modulo a binary clause. In A. Bundy, editor, *12th International Conference on Automated Deduction*, LNCS 814(LNAI), pages 282–295. Springer, 1994.

[23] G. Salzer. Solvable classes of cycle unification problems. In J. Dassow and A. Kelemenova, editors, *Proc. 7th Int. Meeting of Young Computer Scientists (1992)*, Developments in Theoretical Computer Science, pages 215–225. Gordon and Breach Science Publishers, 1994.

[24] M. Schmidt-Schauß. An algorithm for distributive unification. Interner Bericht 13/94, Johann Wolfgang Goethe Universität, Frankfurt, Germany, 1994.

[25] M. Schmidt-Schauß. Unification of stratified second-order terms. Interner Bericht 12/94, Johann Wolfgang Goethe Universität, Frankfurt, Germany, 1994.