Overview of Existing Recurrent Schematizations

Miki Hermann
CRIN (CNRS) and INRIA-Lorraine
Campus Scientifique, BP 239,
54506 Vandœuvre-lès-Nancy, France

Miki.Hermann@loria.fr

Infinite sets of objects with structural similarities appear frequently at different places in the theory of equational reasoning, automatic deduction, logic programming, and databases. The objects in such *infinite families* can have several forms. They are represented most frequently by terms.

The formalisms to deal with infinite families are fairly limited. As long as only the representation is concerned, such formalisms exist and are well-known. The first used formalisms were regular and context-free grammars, later extended by tree automata [GS84]. Recently, this arsenal was enlarged by lazy evaluation and the concept of constraints. The problems appear if we wish to unify infinite families represented by these formalisms. In addition to the drawback that context-free grammars describe only sets of strings, it is undecidable if the intersection of two context-free languages is empty. Hence the unification of context free languages is undecidable since strings in context-free languages are ground terms. Regular grammars and regular tree automata often lack a necessary schematization power. Infinite families often display patterns that cannot be captured by them. Lazy evaluation cannot be combined with unification in a satisfactory way. Constraints represent infinite families implicitly, by restrictions on domains. Therefore, unification of constraint expressions results in constraint solving that can be as problematic as the direct unification of the infinite families itself. Three phenomena are usually observed: (1) the constraint solving is decidable, but the constraint formalism lacks schematization power, (2) the constraint formalism is powerful enough to represent the infinite families but the unification problem for these constraints is undecidable, and (3) the constraint formalism is powerful enough and the unification problem is decidable, but the corresponding constraint solving unification algorithm produces an infinite family of constraints.

In practice, the manipulated sets must be finite, unless there is a possibility to use constraints and implicit representations. The previous considerations clearly indicate the need for a formalism that allows to express explicitly infinite families by finite means, that has a decidable unification problem, and that has a terminating unification algorithm yielding a finite representation of the results. Moreover, such formalism should have a semantics compatible with their domain of application so that it can be easily incorporated into different theoretical developments, as well as into software products.

Schematizations present a suitable formalism to deal explicitly by finite means with infinite families of objets. Under schematization we understand a formalism that effectively describes by finite expressions infinite sets of terms. Contrary to usual constraints that give implicit representations, schematizations must include a mechanism that allows to locate directly every element of a schematized family, they must permit to manipulate effectively the schematized sets, and they must have a finite unification algorithm. Since they provide a different approach to solve the same kind of problems as constraints do, they can be viewed as a new type of constraints.

Different schematizations of first-order terms appeared within the domain of equational reasoning during the last years. Among them, *recurrent schematizations* that attracted a lot of interest in recent years. Roughly speaking, each formalism that belongs to this class schematizes a subclass of primitive recursive functions. Several of these formalisms have been developed as a generalisation of a previous formalism. Hence, there exists a strict hierarchy of recurrent schematizations where always the class of languages schematized by a more recent formalism strictly encompasses the class of languages schematized by an older formalism. The advantage of the recurrent schematizations is the existence of a unification algorithm for them, provided some basic requirements are met. All these formalisms are designed so that the unification problem for the schematized sets is decidable. The known formalisms in this class are: (1) the *ρ-terms* [Che91, CH91, CH95]; (2) the *terms with integer exponents* [Com95]; (3) the *R-terms* [Sal91b, Sal92]; and (4) the *primal grammars* [Her92, Her94, HG97].

The basic idea behind every recurrent schematization is an iteration of contexts. The ρ-terms allow the iteration of only one context in one position. The terms with integer exponents allow the iteration of several contexts, each in one position. The R-terms allow the iteration of several contexts, each in several positions, and also a hierarchy between iterated context. The primal grammars, in addition to the same properties as for R-terms, allow to declare a dependence between the iterations. The unification for all these formalisms is decidable [Her94] and there exists a hierarchy with proper inclusion between these classes of schematized terms [Ama96]. In particular, the terms with integer exponents are more powerful than the ρ-terms, the R-terms are more powerful than the terms with integer exponents, and the primal grammars are slightly more powerful that the R-terms. The ρ-terms and the terms with integer exponents allow to schematize infinite sets of terms with only polynomial growth of symbols, whereas the R-terms and the primal grammars can schematize infinite sets of terms with exponential growth of symbols. Moreover, primal grammars allow the so-called *diagonalisation* that is a source of additional schematization power.

There exist unification algorithms for each of the previously mentioned schematization class. These unification algorithms always terminate and give finite results that represent complete sets of unifiers of the original unification problems. The common feature of all these unification algorithms is a search of a fixpoint in a potentially infinite unification process.

There exist also other recurrent schematizations that were precursors of the previously mentioned schematizations but they either lack expressive power, or are already subsumed by another formalism of the same author, or else their unification problem is undecidable or open. Among them we can find the *recurrent terms* [CHK90], the *R-strings* [Sal91a], the *substitution forms* [Sal91b], and the *primal grammars with marked variables* [Her94, Sal93]. The recurrent terms allow to encode Hibert's 10th problem, therefore their unification is undecidable. The R-string are predecessors of the more general R-terms. In fact, R-string are R-terms restricted to strings. The substitution forms are more general than R-terms but the decidability of their unification problem is open. The undecidability of unification for the primal grammars with marked variables was proved in [Sal93].

References

[Ama96] A. Amaniss. *Méthodes de schématisation pour la démonstration automatique*. PhD thesis, Centre de Recherche en Informatique de Nancy, 1996. To appear.

[CH91] H. Chen and J. Hsiang. Logic programming with recurrence domains. In J. Leach Albert, B. Monien, and M. Rodríguez Artalejo, editors, *Proceedings 18th ICALP Conference, Madrid (Spain)*, volume 510 of *Lecture Notes in Computer Science*, pages 20–34. Springer-Verlag, July 1991.

[CH95] H. Chen and J. Hsiang. Recurrence domains: Their unification and application to logic programming. *Information and Computation*, 122:45–69, 1995.

[Che91] H. Chen. *Automated equational reasoning with structured domains*. PhD thesis, State University of New York at Stony Brook, 1991.

[CHK90] H. Chen, J. Hsiang, and H.-C. Kong. On finite representations of infinite sequences of terms. In S. Kaplan and M. Okada, editors, *Proceedings 2nd International Workshop on Conditional and Typed Rewriting Systems, Montreal (Canada)*, volume 516 of *Lecture Notes in Computer Science*, pages 100–114. Springer-Verlag, June 1990.

[Com95] H. Comon. On unification of terms with integer exponents. *Mathematical Systems Theory*, 28(1):67–88, 1995.

[GS84] M. Gécseg and M. Steinby. *Tree automata*. Akademia kiadó, Budapest, 1984.

[Her92] M. Hermann. On the relation between primitive recursion, schematization, and divergence. In H. Kirchner and G. Levi, editors, *Proceeding 3rd Conference on Algebraic and Logic Programming, Volterra (Italy)*, volume 632 of *Lecture Notes in Computer Science*, pages 115–127. Springer-Verlag, September 1992.

[Her94] M. Hermann. Divergence des systèmes de réécriture et schématisation des ensembles infinis de termes. Habilitation 94-T-065, Université Henri Poincaré Nancy 1, France, March 1994.

[HG97] M. Hermann and R. Galbavý. Unification of infinite sets of terms schematized by primal grammars. *Theoretical Computer Science*, 176(xx):xx–xx, 1997. Accepted for publication. Available by anonymous ftp from `ftp.loria.fr` in the file `pub/loria/eureca/articles/Miki.Hermann/prmgrmunif.ps`.

[OEG91] *Proceedings 7. Österreichische Artificial Intelligence Tagung*, volume 287 of *Informatik-Berichte*. Springer-Verlag, 1991.

[Sal91a] G. Salzer. Deductive generalization and meta-reasoning, or how to formalize Genesis. [OEG91], pages 103–115.

[Sal91b] G. Salzer. *Unification of Meta-Terms*. PhD thesis, Technische Universität Wien, Austria, 1991.

[Sal92] G. Salzer. The unification of infinite sets of terms and its applications. In A. Voronkov, editor, *Proceedings 3rd International Conference on Logic Programming and Automated Reasoning, St. Petersburg (Russia)*, volume 624 of *Lecture Notes in Computer Science (in Artificial Intelligence)*, pages 409–420. Springer-Verlag, July 1992.

[Sal93] G. Salzer. On the relationship between cycle unification and the unification of infinite sets of terms. In W. Snyder, editor, *UNIF '93, Boston (MA, USA)*, June 1993.