Efficient Strategies for Automated Reasoning

Laurent Vigneron *†

The search for definitions of efficient strategies has been for many years a main preoccupation for researchers in automated reasoning for first-order logic. A first step has been done by J.A. Robinson by introducing a proof system for classical logic based on a *resolution* rule. The next step was the addition of a *paramodulation* rule done by G.A. Robinson and Wos. The paramodulation rule permits to deal with the equality predicate without describing explicitly its properties of reflexivity, symmetry, transitivity and functional reflexivity. It also is based on a notion of replacement. Over the time, several refinements have been added to this rule. Brand has shown that only the reflexivity axiom ($x \simeq x$) is needed. Peterson has shown that paramodulations into variables are useless. But the crucial step has been achieved by Hsiang and Rusinowitch [2]. They have introduced ordering restrictions to the application of these rules. Moreover, they have proved the completeness of the *ordering strategy* defined as follows: *each inference step has to be applied between maximal (w.r.t. an ordering) literals in clauses, and in each paramodulation step, a term cannot be replaced by a bigger one.*
Other refinements, such as the superposition strategy which applies replacements only in biggest sides of equations, and clausal simplifications which delete redundant clauses, have followed.

The complete Hsiang-Rusinowitch strategy and others are often unfortunately inefficient in presence of clauses such as the associativity property of an operator f, ($f(f(x,y),z) \simeq f(x,f(y,z))$). It produces the divergence of derivations by successive paramodulations into the subterm $f(x,y)$. Other properties, such as the commutativity of an operator, induce problems with the paramodulation rule because they cannot be oriented.

The most established solution was proposed by Plotkin [5]. He proposed to define an equational theory E, by extracting above properties from the set of clauses, and to define an *unification algorithm modulo* E. This result has been the basis of many works: Lankford and Ballantyne for the particular case of associative and commutative theories, and Peterson and Stickel for completion. When applying ordering strategies to the theorem proving in equational theories, we need to add various additional techniques. The techniques usually proposed in equational deduction are:

- either to add an inference rule applying replacements into axioms of E, and therefore generating *extensions* of these axioms,
- or to associate to each equation the set of its possible extensions, which may be used later by the paramodulation rule.

These extensions have been studied by Jouannaud and H. Kirchner [3], and Bachmair and Dershowitz [1].

We propose here a *new technique* for dealing with these extensions in the clausal frame, by adding two inference rules called *contextual paramodulation* and *extended paramodulation*. In addition, we define an *algorithm* for calculating useful applications of these last

*Department of Computer Science, SUNY at Stony Brook, Stony Brook, NY 11794, USA. E-mail: lavig@cs.sunysb.edu

†Research supported by the *Institut National de Recherche en Informatique et en Automatique*, Rocquencourt (France)

inference rules only by studying theaxioms of E. We prove the refutational completeness of this technique *for any regular theory* E, i.e. any theory whose axioms preserve variables. This generalizes last results of Wertz [7] and Paul [4].

We also propose a *collection of strategies* which decrease the number of possible deductions, and can be combined without loss of completeness.

- *Superposition strategy*: replacements are applied only in biggest sides of equations,
- *Positive strategy* (a variant of the ordering strategy): if a deduction uses a positive literal in a clause, this clause has to be positive, and the positive literal used has to be maximal in the clause; if a deduction uses a negative literal in a clause, this literal has to be maximal w.r.t. the other negative literals of the clause,
- *Simplification rules*: they permit the deletion of redundant clauses.

The above extensions were defined for equational theories. For the case of associative and commutative theories (AC), we define a *constraint strategy* [6]. This extends results of H. & C. Kirchner and Rusinowitch, Nieuwenhuis and Rubio, and Bachmair, Ganzinger, Lynch and Snyder. The principle of our constraint strategy is simple: each unification problem is not solved; it is added to a conjunction of constraints c joined to the deduced clause C. So, a clause $C[\![c]\!]$ represents the set of clauses $C\sigma$ for all substitutions σ solution of c. Considering for instance the unification problem $x*x*x*x*x =^{?}_{AC} y_1*y_2*y_3*y_4$ where $*$ is AC, instead of generating a clause by principal solution (i.e. 34 359 607 481 clauses), we generate only one clause.

Our strategy includes the *basic strategy* of Hullot, whose principle is to forbid replacements into terms introduced by the application of a substitution. We also add non-solved ordering conditions as constraints in the deduced clause. Hence, unification and ordering calculi are transformed into constraint satisfiability checking.

All these results have been implemented in a theorem prover called `DATAC`, for the case of associative and commutative theories. It is an interesting tool for comparing the efficiency of these strategies, and practical results will follow.

References

[1] L. Bachmair and N. Dershowitz. Completion for Rewriting Modulo a Congruence. *Theoretical Computer Science*, 67(2-3):173–202, October 1989.

[2] J. Hsiang and M. Rusinowitch. Proving Refutational Completeness of Theorem Proving Strategies: The Transfinite Semantic Tree Method. *Journal of the ACM*, 38(3):559–587, July 1991.

[3] J.-P. Jouannaud and H. Kirchner. Completion of a Set of Rules Modulo a Set of Equations. *SIAM Journal of Computing*, 15(4):1155–1194, 1986. Preliminary version in Proceedings 11th ACM Symposium on Principles of Programming Languages, Salt Lake City (USA), 1984.

[4] E. Paul (E-mail: etienne.paul@issy.cnet.fr). E-Semantic Tree. Unpublished paper, 70 pages, 1994.

[5] G. Plotkin. Building-in Equational Theories. *Machine Intelligence*, 7:73–90, 1972.

[6] L. Vigneron. Associative-Commutative Deduction with Constraints. In A. Bundy, editor, *Proceedings 12th International Conference on Automated Deduction, Nancy (France)*, volume 814 of *LNAI*, pages 530–544. Springer-Verlag, June 1994.

[7] U. Wertz. First-Order Theorem Proving Modulo Equations. Technical Report MPI-I-92-216, Max Planck Institut für Informatik, April 1992.