Automated Reasoning in Equational Theories

Laurent Vigneron

CRIN-CNRS & INRIA Lorraine
B.P. 239
F-54506 Vandœuvre-lès-Nancy Cedex

E-mail: vigneron@loria.fr

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 [15] 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 [14]. 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 [3] has shown that only the reflexivity axiom ($x = x$) is needed. Peterson has shown that paramodulations into variables are useless. But the crucial step has been achieved by Hsiang and Rusinowitch [4]. 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, $(f(f(x,y),z) = f(x,f(y,z)))$. 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 [13]. 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 [8] for the particular case of associative and commutative theories, and Peterson and Stickel [12] 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 [6], and Bachmair and Dershowitz [1].

We present a 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 inference rules only by studying the axioms of E. We prove the refutational completeness of this technique for a class of equational theories including *any regular theory* [18], i.e. any theory whose axioms preserve variables. This generalizes the last results of Wertz [20] and Paul [11].

We also present a *collection of strategies* decreasing the number of possible deductions, and that 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 strategies*: simplification of clauses and deletion of redundant ones.

The above extensions were defined for equational theories. For the case of associative and commutative theories (AC), we define a *constraint strategy* [10, 17]. This extends results of H. & C. Kirchner and Rusinowitch [7], Nieuwenhuis and Rubio [9], and Bachmair, Ganzinger, Lynch and Snyder [2]. The principle of this constraint strategy is simple: in a deduction step, the unification problem is not solved; it is added as a constraint of the deduced clause. 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 $x * x * x * x =^?_{AC} y_1 * y_2 * y_3 * y_4$ where $*$ is AC, instead of generating one clause per principal solution (i.e. 34 359 607 481 clauses), we generate only one clause.

This constraint strategy includes the *basic strategy* of Hullot [5], 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 [16]. It is an interesting tool for comparing the efficiency of these strategies.
daTac has been applied to study rough equality algebras [19], called R4 and R5, generalizing the well-known S4 and S5 algebras, that are topological Boolean algebras and models for the modal logics S4 and S5, respectively. Rough equality algebras express the notion of uncertainty, but they are so complex it is even very difficult to have an idea of the properties they may satisfy. Such rough algebras are more and more studied in the context of data mining.

References

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

[2] L. Bachmair, H. Ganzinger, C. Lynch, and W. Snyder. Basic paramodulation. *Information and Computation*, 121(2):172–192, 1995.

[3] D. Brand. Proving Theorems with the Modification Method. *SIAM Journal of Computing*, 4:412–430, 1975.

[4] 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.

[5] J.-M. Hullot. *Compilation de Formes Canoniques dans les Théories équationelles.* Thèse de Doctorat de Troisième Cycle, Université de Paris Sud, Orsay (France), 1980.

[6] 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.

[7] C. Kirchner, H. Kirchner, and M. Rusinowitch. Deduction with Symbolic Constraints. *Revue d'Intelligence Artificielle*, 4(3):9–52, 1990. Special issue on Automatic Deduction.

[8] D. S. Lankford and A. Ballantyne. Decision Procedures for Simple Equational Theories with Associative Commutative Axioms: Complete Sets of Associative Commutative Reductions. Technical report, Univ. of Texas at Austin, Dept. of Mathematics and Computer Science, 1977.

[9] R. Nieuwenhuis and A. Rubio. Theorem Proving with Ordering Constrained Clauses. In D. Kapur, editor, *Proceedings 11th International Conference on Automated Deduction, Saratoga Springs (N.Y., USA)*, volume 607 of *Lecture Notes in Computer Science*, pages 477–491. Springer-Verlag, 1992.

[10] R. Nieuwenhuis and A. Rubio. AC-Superposition with Constraints: no AC-unifiers Needed. In A. Bundy, editor, *Proceedings 12th International Conference on Automated Deduction, Nancy (France)*, volume 814 of *Lecture Notes in Artificial Intelligence*, pages 545–559. Springer-Verlag, June 1994.

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

[12] G. E. Peterson and M. E. Stickel. Complete Sets of Reductions for Some Equational Theories. *Journal of the ACM*, 28:233–264, 1981.

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

[14] G. A. Robinson and L. T. Wos. Paramodulation and First-order Theorem Proving. In B. Meltzer and D. Mitchie, editors, *Machine Intelligence*, volume 4, pages 135–150. Edinburgh University Press, 1969.

[15] J. A. Robinson. A Machine-oriented Logic Based on the Resolution Principle. *Journal of the ACM*, 12:23–41, 1965.

[16] M. Rusinowitch and L. Vigneron. Automated Deduction with Associative-Commutative Operators. *Applicable Algebra in Engineering, Communication and Computing*, 6(1):23–56, January 1995.

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

[18] L. Vigneron. Positive Deduction modulo Regular Theories. In H.K. Büning, editor, *Annual Conference of the European Association for Computer Science Logic, Paderborn (Germany)*, volume 1092 of *Lecture Notes in Computer Science*, pages 468–485. Springer-Verlag, September 1995. Selected paper.

[19] L. Vigneron and A. Wasilewska. Modal and Rough Algebras. In P. Borne, editor, *Proceedings of the IMACS/IEEE CESA'96 International Multiconference (Computational Engineering in Systems Applications), Symposium on Modelling, Analysis and Simulation*, pages 1107–1112, Lille (France), July 1996. IMACS/IEEE.

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