Mechanizing Induction for Semantic Data Structures

Adel Bouhoula and Florent Jacquemard

LORIA – INRIA Lorraine
615 Rue du Jardin Botanique – BP 101
54602 Villers-lès-Nancy Cedex, France
Email: {bouhoula, jacquema}@loria.fr
http: //www.loria.fr/{~bouhoula, ~jacquema}

Inductive theorem proving on constrained theories

Data structures like bags, powerlists, square matrices, complete binary trees and sorted lists are ubiquous in software and hardware verifications. These semantic data structures can be easily axiomatized by constrained equational theories. However, little and restrictive work has been carried out on mechanizing induction on such theories since they generate very complex induction schemas.
In this paper, we develop a new approach for solving this problem by adapting and generalizing recent works in tree automata with constraints.

Tree automata with constraints

Tree automata with constraints are used for different purposes in many works on theorem proving by induction. However their role has been somewhat restricted to implicit induction methods, where a proof is driven without explicit induction rules but rather by completion techniques (the term *proofs by consistency* is often used). Lets cite:

- Ground reducibility can be decided with the help of techniques of tree automata with constraints, [6, 5]. This key property expresses that all the ground instance of a given term are reducible by a ground term rewrite system. It is fundamental in automated proofs by consistency in equational theories without constructors [7], and it is also related to *sufficient completeness* in algebraic specifications (see e.g. [8]).
- Comon and Nieuwenhuis [3] make proofs by consistency with saturation methods from first order theorem proving, by the addition to the theory of a first order axiomatization of the minimal Herbrand models. In some cases, such an axiomatization is a tree automata with equality, disequality or inequality constraints, see also [4].
- The first author and Jouannaud [1] have proposed another inductive theorem proving technique for specifications with relations between constructors which are specified by a left-linear and non-conditional rewrite system. In this work, the construction of a tree automaton recognizing irreducible ground terms is involved in order to build a system of free subsorts.

Our procedure

We present here a new procedure for inductive theorem proving on constrained equational theories. This work follows the lines of the test set induction method, used in the prover Spike, see [2]. The general principle in test set induction is to compute first an appropriate explicit induction scheme, called test set, then to generate new conjectures by variable replacement, to simplify these conjectures and apply to normal forms a refutation principle using proof by consistency techniques.

The key idea of our method is to use as an induction scheme a tree automaton with constraints which describes the initial model of the given specification, instead of using test sets.

More precisely, the main steps for an inductive proof with our method are:

1. The computed tree automaton with constraints is used as an induction scheme. This means that it is used like a tree grammar, for the generation of new conjectures.
2. Then, each subgoal is simplified by the axioms and induction hypotheses until we obtain a constructor clause.
3. The clause we obtain is checked to be ground irreducible and valid. In case it is ground irreducible and not valid, then one can terminate the proof by a refutation and the clause is a counter example. Both the validity and ground irreducibility of such clauses are reducible to emptiness tests for the tree automaton with constraints.

Completeness

We also show how to prove completeness of specifications with semantic data structures by generating proof obligations which can be passed over to the inductive prover.

To summarize

Our procedure is sound and refutationally complete.

Moreover, it subsumes all test set induction approaches. It works indeed even when the equational axioms for describing the semantic data structures (constructor axioms) are constrained, non-terminating and non-linear. The other axioms of the specification are constrained Horn clauses with equality.
Based on several examples, this method seems to yield very natural and readable proofs.

Another advantage of our technique is that it allows to handle non terminating rewrite rules modulo equational theories, by applying constrained completion to transform such a theory into an equivalent orientable one (with ordering constraints), see [9]. This allows then to make proofs modulo non orientable axioms, like associativity and commutativity, without having to modify the core of our procedure.

Persepectives

An important step now is to implement this new techniques which we plan to do as part of the Spike system. A big effort would be the development of known emptiness decisions algorithms for subclasses tree automata with equality and disequality constraints. This work also gives us the motivation for the study of new emptiness decision algorithms for extended classes of tree automata with constraints, in particular with ordering constraints.

References

1. A. Bouhoula and J.-P. Jouannaud. Automata-driven automated induction. In *Proceedings 12th IEEE Symposium on Logic in Computer Science, Warsaw (Poland)*, pages 14–25. IEEE Comp. Soc. Press, 1997.
2. Adel Bouhoula and Michaël Rusinowitch. Implicit induction in conditional theories. *Journal of Automated Reasoning*, 14(2):189–235, 1995.
3. H. Comon and R. Nieuwenhuis. Induction = I-axiomatization + first-order consistency. Research Report LSV-98-9, Lab. Specification and Verification, ENS de Cachan, Cachan, France, October 1998. 31 pages.
4. Hubert Comon. Inductive proofs by specifications transformation. In *Proc. 3rd Rewriting Techniques and Applications, Chapel Hill, LNCS 355*, pages 76–91. Springer-Verlag, April 1989.
5. Hubert Comon and Florent Jacquemard. Ground reducibility is exptime-complete. In *Proc. IEEE Symp. on Logic in Computer Science*, Varsaw, June 1997. IEEE Comp. Soc. Press.
6. Dauchet, Caron, and Coquidé. Automata for reduction properties solving. *JSCOMP: Journal of Symbolic Computation*, 20, 1995.
7. Jean-Pierre Jouannaud and Emmanuel Kounalis. Automatic proofs by induction in theories without constructors. *Information and Computation*, 82(1), July 1989.
8. Deepak Kapur, Paliath Narendran, and Hantao Zhang. On sufficient completeness and related properties of term rewriting systems. *Acta Informatica*, 24(4):395–415, 1987.
9. Claude Kirchner, Hélène Kirchner, and M. Rusinowitch. Deduction with symbolic constraints. *Revue d'Intelligence Artificielle*, 4(3):9–52, 1990. Special issue on Automatic Deduction.