Tabled Logic Programming for Verification and Program Analysis

C. R. Ramakrishnan
Department of Computer Science
SUNY at Stony Brook
Stony Brook, NY 11794-4400
e-mail: cram@cs.sunysb.edu

Tabled resolution for logic programs [TS86, CW96] addresses the major shortcomings of Prolog-style resolution, namely, weak termination, repeated subcomputations, and weak semantics for negation. Techniques used for program analysis and verification, such as model checking and abstract interpretation, can be implemented using tabling by treating the semantic equations as logic rules. For instance, we implemented XMC [RRR+97], an efficient model checker for alternation-free modal mu-calculus, by encoding the semantic equations as a logic program, and evaluating the program using the XSB tabled logic programming system [XSB99].

Using the XMC model checker as a starting point, this project aims to:

1. exploit the power of current tabled resolution systems to construct high-performance model checkers for systems specified in various process languages and properties specified using different temporal logics;
2. extend tabling methods to constraint logic programs, thereby tackling new applications, such as verification of infinite-state and real-time systems; and
3. integrate deductive strategies (traditionally used by theorem provers) into model checkers by combining tabled resolution with program transformation techniques.

We elaborate our efforts in each of these directions below.

We have encoded a local model checker for linear-time temporal logic (LTL) as a logic program. A thorough performance evaluation of the LTL model checker is underway. For the mu-calculus model checker, we are investigating optimizations that, based on the given property, reduce the state space of the system being verified. We are also exploring the formulation and implementation of equivalence checking techniques such as symbolic bisimulation checking.

We have implemented XMC/RT, a model checker for timed modal mu-calculus, for verification of real-time systems [DRS99]. For this, we loosely coupled a constraint solver over reals with the XSB tabled logic programming system. This platform yields a model checker for a very large class of properties with performance comparable to several well-known tools for verifying real-time systems. A tighter integration of constraints and tabling is currently underway.

We have taken a significant step towards combining deductive strategies into a model checker. In particular, we have combined induction-based techniques for verification of infinite families of finite state systems (e.g., a n-processor token ring, for n ≥ 2) with model checking. We construct induction proofs using a program transformation framework [RKRR99] we developed recently, and have devised a strategy to guide the search for a proof. This strategy automatically constructs nontrivial induction proofs for verifying safety and liveness properties of several families of systems [RRR+99].

Acknowledgments This work is supported in part by NSF grants CCR-9876242, EIA-9705985, and CCR-9711385.

References