Specification and Evaluation of Logic-Based Model Checking

C. R. Ramakrishnan, I. V. Ramakrishnan, Scott A. Smolka, David S. Warren
Department of Computer Science
SUNY at Stony Brook
Stony Brook, NY 11794-4400
e-mail: {cram,ram,sas,warren}@cs.sunysb.edu
Project Home Page: `http://www.cs.sunysb.edu/~lmc`

The objective of this project is to deploy the latest advances in concurrency research and in logic programming systems to build a software environment for system specification and verification. We are particluarly interested in *model checking*: determining whether a system possesses a formally specified property. The starting points of our research are the Concurrency Factory verification toolkit [CLSS96] and the XSB tabled logic programming system [XSB99].

The first concrete result of our research is XMC, an efficient and flexible model checker for finite-state systems [RRR$^+$97]. XMC is written in under 200 lines of XSB tabled Prolog code, which constitute a declarative specification of CCS and the modal mu-calculus at the level of semantic equations. XMC's encoding centers around two predicates: `trans`, encoding the operational semantics of CCS expressions in terms of a transition relation, and `models`, defining when a CCS expression models a given modal mu-calculus formula. Using tabled resolution to evaluate queries over these predicates yields a *local model checker*, i.e., one that inspects only those states of the system needed to prove or disprove the formula. Despite the high-level nature of XMC's implementation, its performance is comparable to that of highly optimized model checkers such as Spin and Murφ on examples selected from the benchmark suite contained in the standard Spin distribution.

In XMC, systems to be verified are described in XL, a language based on value-passing CCS. We translate XL specifications into compact labeled transition systems using an optimizing compiler which improves verification performance several fold [DR99]. The core principles of this translation have been recently incorporated in Spin, showing similar performance gains. We expect to release the first version of XMC in late 1999.

We have also been actively investigating how XMC's model-checking capabilities can be extended *beyond finite-state systems*. We address this issue with work in two complementary directions: (i) combining constraint solving with tabled resolution, motivated by verification of infinite-state systems; and (ii) developing powerful program-transformation techniques for the verification of infinite families of finite-state systems, i.e. parameterized systems [RKRR99].

Regarding (i), we have built an efficient model checker for real-time systems through the use of a constraint package for the reals on top of tabled resolution. We are currently investigating techniques, based on this implementation, for verification of other kinds of infinite-state systems, such as those involving variables over infinite domains.

Regarding (ii), we have developed an equivalence proof procedure based on logic-program transformations for verifying parameterized systems. We have also formulated a strategy that guides the proof procedure to automatically discover induction-based proofs for properties of various kinds of network topologies. Technical reports describing these investigations, as well as publications on other project-related research, are available from the project home page.

Acknowledgments This work is supported in part by NSF grants EIA-9705998, CCR-9711386, CCR-9805735, and CCR-9876242.

References

[CLSS96] R. Cleaveland, P. M. Lewis, S. A. Smolka, and O. Sokolsky. The Concurrency Factory: A development environment for concurrent systems. In *Computer Aided Verification (CAV)*, 1996.

[DR99] Y. Dong and C.R. Ramakrishnan. An optimizing compiler for efficient model checking. In *Formal Description Techniques For Distributed Systems and Communication Protocols & Protocol Specification, Testing, And Verification (FORTE/PSTV'99)*, 1999.

[RKRR99] A. Roychoudhury, K. Narayan Kumar, C.R. Ramakrishnan, and I.V. Ramakrishnan. A parameterized unfold/fold transformation framework for definite logic programs. In *Principles and Practice of Declarative Programming (PPDP), LNCS 1702*, 1999.

[RRR$^+$97] Y. S. Ramakrishna, C. R. Ramakrishnan, I. V. Ramakrishnan, S. A. Smolka, T. L. Swift, and D. S. Warren. Efficient model checking using tabled resolution. In *Computer Aided Verification (CAV), LNCS 1254*, 1997.

[XSB99] XSB. The XSB logic programming system v2.01, 1999. Available by anonymous ftp from `www.cs.sunysb.edu/~sbprolog`.