

Verification using Tabled Logic Programming^{*}

C. R. Ramakrishnan

`cram@cs.sunysb.edu`

Department of Computer Science, SUNY at Stony Brook
Stony Brook, NY 11794-4400, USA

Abstract

The LMC project aims to advance the state of the art of system specification and verification using the latest developments in logic programming technology [CDD⁺98]. Initially, the project was focussed on developing an efficient model checker, called XMC [RRR⁺97], for value-passing CCS [Mil89] and the modal mu-calculus [Koz83] based on the XSB logic programming system [XSB00]. We developed an optimizing compiler to translate specifications in a dialect of value-passing CCS to compact labeled transition systems [DR99], improving verification performance several fold. The core principles of this translation have been recently incorporated in SPIN [Hol97] showing similar gains in performance [Hol99]. The XMC system can be downloaded from <http://www.cs.sunysb.edu/~lmc>.

More recently, we have developed

- techniques using logic-program transformations [TS84,PP99,RKRR99] for verifying parameterized systems, i.e., infinite families of finite-state systems [RKR⁺00];
- a proof-tree viewer for justifying successful or failed verification runs for branching-time properties [RRR00];
- a symbolic bisimulation checker (based on the work of [HL95]) for value-passing systems [MRRV00];
- model checkers for
 - real-time systems [DRS00] based loosely on the local model checking algorithm of [SS95];
 - LTL with actions [PR00] based on GCTL* of [BCG00] and the on-the-fly model checking algorithm in [BCG95];

In this tutorial, we describe the XMC system as well as the above developments. In addition, we outline the research efforts of the verification and the logic programming community that have been instrumental in these developments.

^{*} Research supported in part by NSF grants EIA-9705998, CCR-9711386, CCR-9805735, and CCR-9876242.

References

- [BCG95] G. Bhat, R. Cleaveland, and O. Grumberg. Efficient on-the-fly model checking for CTL*. In *IEEE Symposium on Logic in Computer Science*. IEEE Press, 1995.
- [BCG00] G. Bhat, R. Cleaveland, and A. Groce. Efficient model checking via Büchi tableau automata. Technical report, Department of Computer Science, SUNY, Stony Brook, 2000.
- [CDD⁺98] B. Cui, Y. Dong, X. Du, K. Narayan Kumar, C. R. Ramakrishnan, I. V. Ramakrishnan, A. Roychoudhury, S. A. Smolka, and D. S. Warren. Logic programming and model checking. In *Static Analysis Symposium*. Springer Verlag, 1998.
- [DR99] Y. Dong and C. R. Ramakrishnan. An optimizing compiler for efficient model checking. In *Proceedings of FORTE/PSTV '99*, 1999.
- [DRS00] X. Du, C. R. Ramakrishnan, and S. A. Smolka. Tabled resolution + constraints: A recipe for model checking real-time systems. Technical report, Department of Computer Science, SUNY, Stony Brook, 2000. URL: <http://www.cs.sunysb.edu/~cram/papers>.
- [HL95] M. Hennessy and H. Lin. Symbolic bisimulations. *Theoretical Computer Science*, 138:353–389, 1995.
- [Hol97] G. J. Holzmann. The model checker SPIN. *IEEE Transactions on Software Engineering*, 23(5):279–295, May 1997.
- [Hol99] G. Holzmann. The engineering of a model checker: Gnu i-protocol case study revisited. In *6th SPIN Workshop on Practical Aspects of Model Checking*, 1999.
- [Koz83] D. Kozen. Results on the propositional μ -calculus. *Theoretical Computer Science*, 27:333–354, 1983.
- [Mil89] R. Milner. *Communication and Concurrency*. International Series in Computer Science. Prentice Hall, 1989.
- [MRRV00] M. Mukund, C. R. Ramakrishnan, I. V. Ramakrishnan, and R. Verma. Symbolic bisimulation using tabled constraint logic programming. Technical report, Department of Computer Science, SUNY, Stony Brook, 2000. URL: <http://www.cs.sunysb.edu/~cram/papers>.
- [PP99] A. Pettorossi and M. Proietti. Synthesis and transformation of logic programs using unfold/fold proofs. *Journal of Logic Programming*, 1999.
- [PR00] L. R. Pokorny and C. R. Ramakrishnan. Model checking linear temporal logic using tabled logic programming. Technical report, Department of Computer Science, SUNY, Stony Brook, 2000. URL: <http://www.cs.sunysb.edu/~cram/papers>.
- [RKR⁺00] A. Roychoudhury, K. Narayan Kumar, C. R. Ramakrishnan, I. V. Ramakrishnan, and S. A. Smolka. Verification of parameterized systems using logic-program transformations. In *Proceedings of TACAS 2000*. Springer-Verlag, 2000.
- [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)*, volume 1702 of *Lecture Notes in Computer Science*, pages 396–413, 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 *Proceedings of the 9th International Conference on Computer-Aided Verification (CAV '97)*, Haifa, Israel, July 1997. Springer-Verlag.
- [RRR00] A. Roychoudhury, C. R. Ramakrishnan, and I. V. Ramakrishnan. Justifying proofs using memo tables. In *ACM Conference on Principles and Practice of Declarative Programming (PPDP)*, 2000.
- [SS95] O. Sokolsky and S. A. Smolka. Local model checking for real-time systems. In *Proceedings of the 7th International Conference on Computer-Aided Verification*. American Mathematical Society, 1995.
- [TS84] H. Tamaki and T. Sato. Fold/unfold transformation of logic programs. In *International Conference on Logic Programming*, pages 127–138, 1984.
- [XSB00] The XSB Group. The XSB logic programming system v2.1, 2000. Available from <http://www.cs.sunysb.edu/~sbprolog>.