Extensible Symbolic System Analysis*

José Meseguer

University of Illinois at Urbana-Champaign, USA

Abstract

Unification and narrowing are a key ingredient not only to solve equations modulo an equational theory, but also to perform symbolic system analysis. The key idea is that a concurrent system can be naturally specified as a rewrite theory $\mathcal{R} = (\Sigma, E, R)$, where (Σ, E) is an equational theory specifying the system's states as an algebraic data type, and R specifies the system's concurrent, and often non-deterministic, transitions. One can perform such symbolic analysis by describing sets of states as (the instances of) terms with logical variables, and using narrowing *modulo* E to symbolically perform transitions. Under reasonable conditions on $\mathcal{R}$, this idea can be applied not only for complete reachability analysis, but also for temporal logic model checking. This approach is quite flexible but has some limitations. Could it be possible to make symbolic system analysis techniques more *extensible* and more widely applicable by simultaneously combining the powers of rewriting, narrowing, SMT solving and model checking? We give a progress report on current work aiming at such a unified symbolic approach.

1 Introduction

The automatic analysis of systems through model checking is one of the most successful system verification methods. The standard approaches (see, e.g., [13]) assume a finite-state system whose state space is exhaustively explored to check whether a system satisfies a desired temporal logic property. However, systems are often *infinite-state* in two possible ways (or are simultaneously infinite in these two ways):

1. The number of *initial states* is infinite, even though the set of states reachable from each initial state may be finite. For example, systems *parametric* in the number of processes or objects are of this kind.

2. The number of states *reachable* from an initial state is infinite. This often happens because states contain *unbounded data structures*.

To cope with these two sources of infinity two complementary methods can be used. On the one hand, *state abstraction* and *parametric system* techniques (see, e.g., [13]) can reduce the verification of infinite-state systems to that of finite-state ones. One the other hand, *infinite-state model checking* methods can be used, based on various kinds of *symbolic techniques* such as: (i) automata and grammars, e.g., [1, 12, 10, 11, 20, 23, 4, 3, 2]; (ii) SMT solving, e.g., [5, 14, 18, 19, 22, 26, 27]; and (iii) narrowing [25, 16, 17, 8, 9, 7].

We can think of these various infinite-state symbolic analysis techniques a *niches*, so that: (i) if a system specification can be cast within one of them, and (ii) if the chosen symbolic method can deal with the temporal logic property of interest (some methods only support *reachability analysis*, not general temporal logic model checking), then symbolic analysis is possible.

A key open research issue limiting the applicability of current symbolic techniques is lack of, or limited support for, *extensibility*. That is, although certain classes of systems can be

*Research partially supported by NSF Grant CNS 13-19109.

formalized in ways that allow the application of specific symbolic analysis techniques, many other systems of interest fall outside the scope of some existing symbolic techniques. In such cases one would like to *extend and combine* the power of symbolic techniques to analyze the given system. Indeed, it seems fair to say that at present we lack *general extensibility techniques for symbolic analysis* that can *simultaneously* combine the power of SMT solving, rewriting- and unification-based analysis, and automata-based model checking; and we lack tools that can apply them *together* to analyze a wide variety of systems beyond the scope of each separate analysis technique.

2 Towards Extensible Symbolic System Analysis

Several of us at the University of Illinois at Urbana-Champaign, SRI International, the Universitat Politécnica de of Valéncia, the Escuela Colombiana de Ingeniería, NASA Langley, the Naval Research Laboratory, and the University of Waterloo in Canada (more on this in the Acknowledgments) are currently working on developing the foundations and implementations of techniques that can simultaneously support symbolic analysis using SMT solving, rewriting/narrowing methods, and automata-based model checking.

More precisely, a concurrent system can be naturally specified as a *rewrite theory* [21] $\mathcal{R} = (\Sigma, E_0 \cup E, R)$ where: (i) $(\Sigma, E_0 \cup E)$ is an equational theory describing the system's states as an algebraic data type; and (ii) R is a collection of rewrite rules specifying the *system transitions*. Furthermore, we can often identify an equational subtheory $(\Sigma_0, E_0) \subseteq (\Sigma, E_0 \cup E)$ such that initial algebra $\mathcal{T}_{\Sigma_0/E_0}$ of (Σ_0, E_0) has a *decidable* first-order theory, whose satisfiability can be decided by an SMT solver, and, furthermore, the subtheory $(\Sigma_0, E_0) \subseteq (\Sigma, E_0 \cup E)$ is *protected* by the inclusion (i.e., we have an isomorphism $\mathcal{T}_{\Sigma_0/E_0} \cong \mathcal{T}_{\Sigma/E_0 \cup E|\Sigma_0}$). The extensible symbolic methods sketched above are methods to reason symbolically about the initial model $\mathcal{T}_{\mathcal{R}}$ of the rewrite theory $\mathcal{R}$, which in general may be the model of an infinite-state system.

The technical steps we are taking to achieve the goal of extensible symbolic analysis can be visualized, and placed in the context of existing work, by considering Figure 1 below.

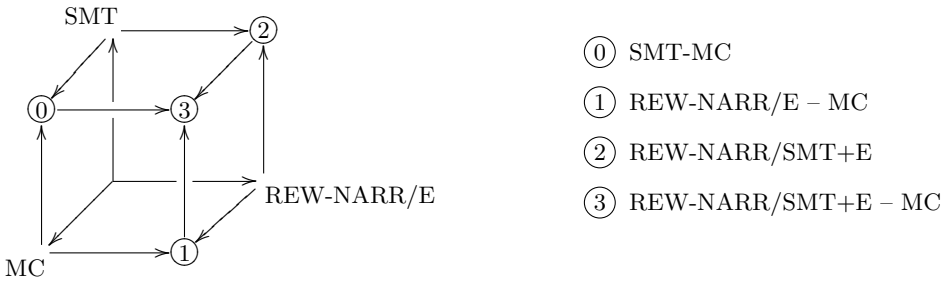

Figure 1: Combining techniques for extensible symbolic analysis

The three separate thrusts of symbolic analysis already mentioned, namely: (i) SMT solving, (ii) rewriting and unification-based techniques (modulo a theory E), and (iii) automata-based model checking are respectively abbreviated as the SMT, REW-NARR/E, and MC vectors in the cube. Vertices 0, 1, and 2 describe *pairwise combinations* obtainable as endpoints of vector additions for two of these three basic vectors. For example, vertex 0 describes SMT-based model checking, which is a very active area of research (see, e.g., [5, 14, 18, 19, 22, 26, 27]). Vertex 1 includes work on both rewriting-based model checking, e.g., [15, 6], and narrowing-based symbolic model checking , e.g., [25, 16, 17, 8, 9, 7]. Vertex 3 is the endpoint of adding the three

basic vectors, so that the joint power of the three symbolic analysis methods can be brought to bear on a much broader class of systems. A first, partial step towards reaching Vertex 3 is model checking based on *rewriting modulo SMT* [24], but the full power should be achieved through *narrowing modulo SMT* techniques currently under development.

Although such a combination of symbolic methods should make the analysis of systems much more extensible, there is already ample evidence from the work on narrowing-based model checking suggesting that symbolic techniques should be used in tandem with abstraction and other space state reduction techniques, which often remain necessary —or are in any case very useful even when not strictly needed— to make model checking decidable [17, 16, 8, 9, 7].

Acknowledgments

The development and implementation of these ideas, many of them ongoing, is joint work with various colleagues and Ph.D. students, including: Kyungmin Bae, Andrew Cholewa, Santiago Escobar, Steven Eker, Vijay Ganesh, Catherine Meadows, César Muñoz, Camilo Rocha, and Carolyn Talcott.

References

[1] P. Abdulla, B. Jonsson, P. Mahata, and J. dOrso. Regular tree model checking. In *Computer Aided Verification*, pages 452–466. Springer, 2002.

[2] R. Alur, C. Courcoubetis, T. A. Henzinger, and P.-H. Ho. Hybrid automata: an algorithmic approach to the specification and verification of hybrid systems. In R.L. Grossman, A. Nerode, A.P. Ravn, and H. Rischel, editors, *Workshop on Theory of Hybrid Systems*, pages 209–229. Springer LNCS 739, 1993.

[3] R. Alur and D. L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126(2):183–235, 1994.

[4] Rajeev Alur and P. Madhusudan. Adding nesting structure to words. *J. ACM*, 56(3), 2009.

[5] A. Armando, J. Mantovani, and L. Platania. Bounded model checking of software using SMT solvers instead of SAT solvers. *Model Checking Software*, pages 146–162, 2006.

[6] K. Bae and J. Meseguer. Model checking linear temporal logic of rewriting formulas under localized fairness. To appear in *Science of Computer Programming*, 2014.

[7] K. Bae and J. Meseguer. Predicate abstraction of rewrite theories. To appear in Proc. *RTA 2014*, Springer LNCS, 2014.

[8] Kyungmin Bae, Santiago Escobar, and Jose Meseguer. Abstract Logical Model Checking of Infinite-State Systems Using Narrowing. In *Rewriting Techniques and Applications (RTA'13)*, volume 21 of *LIPIcs*, pages 81–96. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2013.

[9] Kyungmin Bae, Santiago Escobar, and José Meseguer. Infinite-State Model Checking of LTLR Formulas Using Narrowing. In *Proc. WRLA 2014*. Springer LNCS, to appear, 2014.

[10] A. Bouajjani. Languages, rewriting systems, and verification of infinite-state systems. *Automata, Languages and Programming*, pages 24–39, 2001.

[11] A. Bouajjani and J. Esparza. Rewriting models of boolean programs. *Term Rewriting and Applications*, pages 136–150, 2006.

[12] A. Bouajjani, B. Jonsson, M. Nilsson, and T. Touili. Regular model checking. In *Computer Aided Verification*, pages 403–418. Springer, 2000.

[13] Edmund M. Clarke, Orna. Grumberg, and Doron A. Peled. *Model Checking*. MIT Press, 2001.

[14] L. Cordeiro, B. Fischer, and J. Marques-Silva. SMT-based bounded model checking for embedded ansi-c software. In *ASE*, pages 137–148. IEEE, 2009.

[15] Steven Eker, José Meseguer, and Ambarish Sridharanarayanan. The Maude LTL model checker. In F. Gadducci and U. Montanari, editors, *Proc. 4th. Intl. Workshop on Rewriting Logic and its Applications*. ENTCS, Elsevier, 2002.

[16] Santiago Escobar, Catherine Meadows, and José Meseguer. Maude-npa: Cryptographic protocol analysis modulo equational properties. In *Foundations of Security Analysis and Design V, FOSAD 2007/2008/2009 Tutorial Lectures*, volume 5705 of *Lecture Notes in Computer Science*, pages 1–50. Springer, 2009.

[17] Santiago Escobar and José Meseguer. Symbolic model checking of infinite-state systems using narrowing. In *RTA*, volume 4533 of *Lecture Notes in Computer Science*, pages 153–168, 2007.

[18] M.K. Ganai and A. Gupta. Accelerating high-level bounded model checking. In *ICCAD*, pages 794–801. ACM, 2006.

[19] M.K. Ganai and A. Gupta. Completeness in SMT-based BMC for software programs. In *DATE*, pages 831–836. IEEE, 2008.

[20] T. Genet and V. Tong. Reachability analysis of term rewriting systems with timbuk. In *Logic for Programming, Artificial Intelligence, and Reasoning*, pages 695–706. Springer, 2001.

[21] José Meseguer. Conditional rewriting logic as a unified model of concurrency. *Theoretical Computer Science*, 96(1):73–155, 1992.

[22] A. Milicevic and H. Kugler. Model checking using SMT and theory of lists. *NASA Formal Methods*, pages 282–297, 2011.

[23] H. Ohsaki, H. Seki, and T. Takai. Recognizing boolean closed a-tree languages with membership conditional rewriting mechanism. In *Rewriting Techniques and Applications*, pages 483–498. Springer, 2003.

[24] Camilo Rocha, José Meseguer, and César Mu noz. Rewriting Modulo SMT and Open System Analysis. In *Proc. WRLA 2014*. Springer LNCS, to appear, 2014.

[25] P. Thati and J. Meseguer. Symbolic reachability analysis using narrowing and its application to the verification of cryptographic protocols. *J. Higher-Order and Symbolic Computation*, 20(1–2):123–160, 2007.

[26] M. Veanes, N. Bjørner, and A. Raschke. An SMT approach to bounded reachability analysis of model programs. In *FORTE*, pages 53–68. Springer, 2008.

[27] D. Walter, S. Little, and C. Myers. Bounded model checking of analog and mixed-signal circuits using an SMT solver. *Automated Technology for Verification and Analysis*, pages 66–81, 2007.