Symbolic state space generation for finite models with unknown bounds

Gianfranco Ciardo Robert Marmorstein Radu Siminiceanu

College of William and Mary, Williamsburg, Virginia 23187
{ciardo,rmmarm,radu}@cs.wm.edu

Abstract

In previous work, we proposed a "saturation" algorithm for symbolic state-space generation based on multi-valued decision diagrams and boolean Kronecker operators. This approach greatly outperforms traditional BDD-based techniques but, like them, assumes a priori knowledge of the state space of each submodel. In this work, we introduce a new algorithm that merges explicit local state-space discovery with symbolic global state-space generation. This relieves the modeler from worrying about the behavior of submodels in isolation.

1 Introduction

Since the introduction of implicit methods in verification and symbolic model checking, decision diagram, in particular BDDs [1, 2, 3], have had enormous success. However, the systems targeted have been mainly VLSI and protocols, where the possible values of the state variable is easily determined a priori. When attempting to use these same techniques to verify arbitrary systems modeled in a high-level specification language such as Petri nets or pseudocode, determining the range of the state variables is much more difficult. Traditionally, the user has had to deal with this difficulty. In NuSMV [6], the domain for multi-valued variables must be given explicitly as a set or integer range. In our own work, the input (a Petri net) must be partitioned so that the state space of each resulting subnet can be generated in isolation; in practice, the modeler is forced to carefully add inhibitor arcs or other constraints to the net to realize this property. Putting this burden in the user is limiting and possibly error-prone.

We tackle this problem with an algorithm that, with minimal overhead, merges explicit exploration of the local state space of each submodel with symbolic exploration of the global state space to produce a multi-valued decision diagram (MDD) [7] representation of the final state-space, together with the exact representation of the local sub-spaces. The algorithm is based on our efficient

saturation algorithm [5], which uses an MDD to store the states and boolean Kronecker matrices to encode the transition relation. By breaking the traditionally monolithic transition function into event-based functions, exploiting *event locality* [4], and performing *in-place updates* of MDD nodes, the saturation algorithm showed massive time and space improvements over previously-known approaches. Our new algorithm also employs these ideas and is more general, as it can be applied to models where the traditional algorithm would fail.

References

[1] R. E. Bryant. Graph-based algorithms for boolean function manipulation. *IEEE Trans. Comp.*, 35(8):677–691, Aug. 1986.

[2] R. E. Bryant. Symbolic boolean manipulation with ordered binary-decision diagrams. *ACM Comp. Surv.*, 24(3):293–318, 1992.

[3] J. R. Burch, E. M. Clarke, K. L. McMillan, D. L. Dilll, and L. J. Hwang. Symbolic model checking: 10^{20} states and beyond. *Information and Computation*, 98:142–170, 1992.

[4] G. Ciardo, G. Luettgen, and R. Siminiceanu. Efficient symbolic state-space construction for asynchronous systems. In M. Nielsen and D. Simpson, editors, *Proc. 21th Int. Conf. on Applications and Theory of Petri Nets*, LNCS 1825, pages 103–122, Aarhus, Denmark, June 2000. Springer-Verlag.

[5] G. Ciardo, G. Luettgen, and R. Siminiceanu. Saturation: An efficient iteration strategy for symbolic state space generation. In T. Margaria and W. Yi, editors, *Proc. Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, LNCS 2031, pages 328–342, Genova, Italy, Apr. 2001. Springer-Verlag.

[6] A. Cimatti, E. Clarke, F. Giunchiglia, and M. Roveri. NuSMV: A new symbolic model verifier. In N. Halbwachs and D. Peled, editors, *CAV'99*, LNCS 1633, pages 495–499, Trento, Italy, 1999. Springer-Verlag.

[7] T. Kam, T. Villa, R. Brayton, and A. Sangiovanni-Vincentelli. Multi-valued decision diagrams: theory and applications. *Multiple-Valued Logic*, 4(1–2):9–62, 1998.