SAT-Based Model Checking: Interpolation, IC3 and Beyond

Orna Grumberg
The Technion, Haifa, Israel

Model checking [3] is an automatic approach to formally verifying that a given system satisfies a given specification. The system to be verified is modelled as a finite state machine and the specification is described using temporal logic. Model checking algorithms are typically based on an exploration of the model state space while searching for violations of the specification. In spite of its great success in verifying hardware and software systems, the applicability of model checking is impeded by its high space and time requirements. This is referred to as the *state explosion problem.*

The introduction of SAT-based model checking algorithms [1,8,6,9,2] significantly increases the size of the systems that can be model checked. In its early days SATbased model checking was used mostly for bug hunting. The introduction of *interpolation* enabled an efficient complete algorithm, referred to as Interpolation-based model checking (ITP) [6]. ITP uses interpolation to extract an over-approximation of a set of reachable states from a proof of unsatisfiability generated by a SAT-solver. The set of reachable states computed by the reachability analysis is used by ITP to check if a system M satisfies a safety property AGp.

In [2] an alternative SAT-based algorithm, called IC3, is introduced. Similarly to ITP, IC3 also computes over-approximations of sets of reachable states. However, ITP unrolls the model in order to obtain more precise approximations. In many cases, this is a bottleneck of the approach. IC3, on the other hand, improves the precision of the approximations by performing many local checks that do not require unrolling.

Here, we survey several approaches to enhancing SAT-based model checking. One approach, detailed in [9], uses *interpolation sequence* [5,7] rather than interpolation in order to obtain a more precise over-approximation of the set of reachable states.

Another approach [11], based on interpolation, extracts interpolants in both forward and backward manner and exploits them for an *intertwined approximated forward and backward reachability analysis.* This approach is mostly local and avoids unrolling of the checked model as much as possible.

A different approach, described in [10], integrates lazy abstraction with IC3 in order to achieve scalability. *Lazy abstraction* [4,7], originally developed for software model checking, is a specific type of abstraction that allows hiding different model details at different steps of the verification. We find the IC3 algorithm most suitable for lazy abstraction since its state traversal is performed by means of local reachability checks, each involving only two consecutive sets. A different abstraction can therefore be applied in each of the local checks.

All these approaches result in efficient and complete SAT-based verification algorithms.

References

[1] A. Biere, A. Cimatt, E. Clarke, Y. Zhu. *Symbolic Model Checking Without BDDs* In: TACAS, 1999.

[2] A. R. Bradley. *SAT-based Model Checking without Unrolling* In: VMCAI, 2011.

[3] E. C. Clarke, O. Grumberg, D. Peled. *Model Checking.* MIT Press, 1999.

[4] T. Henzinger, R. M. R. Jhala. *Lazy Abstraction.* In: POPL, 2002.

[5] R. Jhala, K. McMillan. *Interpolant-based transition relation approximation.* In: CAV, 2005.

[6] K. L. McMillan. *Interpolation and SAT-based Model Checking.* In: CAV, 2003.

[7] K. L. McMillan. *Lazy Abstraction with Interpolants.* In: CAV, 2006.

[8] M. Sheeran, S. Singh, G. Stålmarck. *Checking Safety Properties using Induction and a SAT-solver.* In: FMCAD, 2000.

[9] Y. Vizel, O. Grumberg. *Interpolation-sequence based Model Checking.* In: FMCAD, 2009.

[10] Y. Vizel, O. Grumberg, S. Shoham. *Lazy Abstraction and SAT-based Reachability in Hardware Model Checking.* In: FMCAD, 2012.

[11] Y. Vizel, O. Grumberg, S. Shoham. *Intertwined Forward-backward Reachability Analysis using Interpolants.* In: TACAS'13, LNCS 7795, pp. 308-323, Springer, 2013.