Bounded Model Checking

Andrzej Zbrzezny
a.zbrzezny@ajd.czest.pl

Institute of Mathematics and Computer Science
Jan Długosz University in Częstochowa

In this tutorial, I will present the main concepts, algorithms, and tools of bounded model checking.

Model checking is an automatic verifying technique for finite state concurrent systems and concurrent systems that have finite-state abstractions. In order to check automatically whether the system satisfies a given property, one must first create a model of the system and then describe in a formal language both the created model and the property. A prerequisite for model checking is a model of the system under consideration. A standard class of models to represent software and hardware systems are transitions systems.

A transition system is a tuple $(S, Act,\ TR, I, AP, L)$ consisting of a set of states S, a set of actions Act, a transition relation $\longrightarrow \subseteq S \times Act \times S$, a set of initial states $I \subseteq S$, a set of atomic propositions AP, and a labelling function $L : S \rightarrow 2^{AP}$.

To formulate properties of the systems suitable temporal logics are in use. The most common used are linear temporal logic, computation tree logic, a full branching time logic, the universal and existential fragments of these logics, and other logics which are their adaptations and extensions.

The practical applicability of model checking is strongly limited by the state explosion problem, which means that the number of model states grows exponentially in the size of the system representation. A number of state reduction techniques and symbolic model checking approaches are in use to avoid this problem.

Bounded model checking (BMC) is one of symbolic model checking methods. SAT-based BMC reduces the problem of truth of a temporal formula in a transition system to the problem of satisfiability of a formula of the classical propositional calculus. SMT-based BMC reduces the problem of truth of a temporal formula in a transition system to the problem of satisfiability of a quantifier-free first-order formula. Both the methods translate the transition relation and a given property to a suitable formula.

For a given temporal logic TL the application of the BMC method requires proving the theorem which provides the basis for the verification of formulae of this logic in a given transition system using finite prefixes of paths. A finite prefix of the length $k \geqslant 0$ of a path is called a k-path. The theorem in question says that a formula φ of temporal logic TL is true in a transition system $\mathcal{M}$ if and only if there exists a natural number k, such that the formula $[\mathcal{M}, \varphi]_k$ being a conjunction of a formula encoding a finite set of k-paths and a formula being a translation of the formula φ, is satisfiable.

The theorem above justifies the correctness of the standard BMC algorithm. Starting with $k = 0$, the algorithm creates, for a given transition system $\mathcal{M}$ and a given formula φ, a formula $[\mathcal{M}, \varphi]_k$. Then the formula $[\mathcal{M}, \varphi]_k$ is converted to a satisfiability-equivalent propositional formula in conjunctive normal form and forwarded to either a SAT solver or SMT solver. If the tested formula is not satisfiable, then k is increased (usually by 1) and the process is repeated.

The BMC algorithm terminates if either the formula $[\mathcal{M}, \varphi]_k$ turns out to be satisfiable for some k, or k becomes greater than a certain, $\mathcal{M}$-dependent threshold (e.g. the number of states of $\mathcal{M}$). Exceeding this threshold means that the formula φ is not true in the transition system $\mathcal{M}$. On the other hand, satisfiability of $[\mathcal{M}, \varphi]_k$, for some k means that the formula φ is true in the transition system $\mathcal{M}$. Moreover, the valuation found by a (SAT or SMT) solver allows to determine a set of k-paths, which is a witness for φ.

The BMC algorithm also terminates if, for some k, the available resources (memory and/or time) are insufficient either to generate the formula $[\mathcal{M}, \varphi]_k$ or to check its satisfiability by the solver. In such a case, it means that the BMC algorithm is not able to check whether the property expressed by the formula φ holds in the transition system $\mathcal{M}$ due to limited resources available.

In the last part of the lecture, I will show the application of bounded model checking to selected concurrent systems.