Integrating BDD-Based and SAT-Based Symbolic Model Checking

Alessandro Cimatti¹, Enrico Giunchiglia², Marco Pistore¹, Marco Roveri¹, Roberto Sebastiani³, and Armando Tacchella²

¹ ITC-IRST, Via Sommarive 18, 38050 Trento, Italy
{cimatti,pistore,roveri}@irst.itc.it
² DIST – Università di Genova, Viale Causa 13, 16145 Genova, Italy
{enrico,tac}@mrg.dist.unige.it
³ Università di Trento, Via Sommarive 14, 38050 Trento, Italy
rseba@science.unitn.it

Abstract. Symbolic model checking is a very successful formal verification technique, classically based on Binary Decision Diagrams (BDDs). Recently, propositional satisfiability (SAT) techniques have been proposed as a computational basis for symbolic model checking, and proved to be an effective alternative to BDD-based techniques. In this paper we show how BDD-based and SAT-based techniques have been effectively integrated within the NuSMV symbolic model checker.

1 Introduction

Model checking [11,20] is a formal technique for the verification of finite state systems. The system being analyzed is represented as a Finite State Machine (FSM), while the requirements to be satisfied are expressed in temporal logics, e.g. Computation Tree Logic (CTL), or Linear Temporal Logic (LTL). Model checking algorithms are based on the exhaustive analysis of the state space of the FSM. They are able to prove that the system satisfies the requirement, or, more importantly, are able to produce a counterexample, i.e. a behaviour of the FSM that violates the requirements. Model Checking is an extremely effective debugging technique, and is being applied in several application domains, ranging from the analysis of telecommunication protocols to reactive controllers to hardware designs.

Originally, model checking was implemented by means of “explicit-state” techniques, where single states of the FSM are analyzed and stored. One of the most notable examples of explicit-state model checking is SPIN [17], that is very effective in the analysis of asynchronous systems. In general, for many application domains, the large amount of computational resources needed to analyze real-size designs (the so-called state-explosion problem) may be a significant limitation. The introduction of Symbolic Model Checking [18] made it possible to explore state spaces of extremely large size. In symbolic model checking, instead of manipulating individual states, the algorithms manipulate sets of states. These are compactly represented and efficiently constructed by means of Binary Decision

Diagrams \([6]\) (BDDs), that are canonical forms for propositional formulae. Since the seminal work of McMillan \([18]\), several mechanisms for a partitioned representation of finite state machines and different exploration styles \([7,22,13]\) have allowed to increase the applicability of BDD-based model checking. Recently, a new form of symbolic model checking, commonly known as Bounded Model Checking \([4]\), has been introduced. Bounded Model Checking is based on the encoding of a model checking problem into a propositional satisfiability (SAT) problem, and on the application of efficient SAT solvers. This approach, in the following called SAT-based model checking, relies on the enormous progress in the field of propositional satisfiability \([14]\). The approach is currently enjoying a substantial success in several industrial fields (see, e.g., \([12]\), but also \([5]\)), and opens up new research directions.

BDD-based and SAT-based model checking are often able to solve different classes of problems, and can therefore be seen as complementary techniques. The effective integration of BDD-based and SAT-based model checking techniques is very important to widen the spectrum of applicability of symbolic model checkers. Goal of this paper is to describe how the BDD-based and SAT-based approaches to symbolic model checking have been successfully integrated within the NuSMV model checker. In Section 2 we outline the NuSMV project. In Section 3 and 4 we describe the functionalities and the architecture of NuSMV. In Section 5 we discuss some results and outline directions for future development.

2 The NuSMV Symbolic Model Checker

NuSMV is a symbolic model checker originated from the reengineering, reimplementation and extension of SMV \([18]\), the original BDD-based model checker developed by McMillan et al. at CMU (SMV from now on). The NuSMV project aims at the development of a state-of-the-art symbolic model checker, designed to be applicable in technology transfer projects: it is a well structured, open, flexible and documented platform for model checking, and is robust and close to industrial systems standards \([8]\).

The first version of NuSMV, called NuSMV1 in the following, basically implements BDD-based symbolic model checking. The second version of NuSMV (NuSMV2 in the following), inherits all the functionalities and the implementation style of the previous version. However, NuSMV2 significantly extends the functionalities of NuSMV1, and its internal structure departs from the one of NuSMV1. The main novelty in NuSMV2 is the integration of model checking techniques based on propositional satisfiability. Remarkably, the integration covers the whole input language of NuSMV. NuSMV2 is currently the only publicly available system that allows for both BDD-based and SAT-based model checking. In order to integrate SAT-based and BDD-based model checking, a major architectural redesign was carried out in NuSMV2, in order to make as many functionalities as possible independent of the actual model checking engine used. An example of this are the services provided by the modules implementing