Chapter 8
BDD-Based Symbolic Model Checking

Sagar Chaki and Arie Gurfinkel

Abstract Symbolic model checking based on Binary Decision Diagrams (BDDs) is one of the most celebrated breakthroughs in the area of formal verification. It was originally proposed in the context of hardware model checking, and advanced the state of the art in model-checking capability by several orders of magnitude in terms of the sizes of state spaces that could be explored successfully. More recently, it has been extended to the domain of software verification as well, and several BDD-based model checkers for Boolean programs and push-down systems have been developed. In this chapter, we summarize some of the key concepts and techniques that have emerged in this story of successful practical verification.

8.1 Introduction

Algorithms for temporal logic model checking [14] were initially implemented in an explicit-state manner. This means that all automata involved in verification were represented using explicit graph-based data structures. Such automata include the Kripke structures as well as Büchi automata and tableaux obtained from the temporal logic specifications. In particular, the edges of the graph (which are in the worst case quadratic in the number of nodes) were represented using adjacency lists, matrices, etc.

From a theoretical perspective, the data structure used to represent the automata makes no difference whatsoever. From a practical perspective, however, this meant that model checkers could only handle automata with at most 10^3 to 10^6 reachable states [8]. Verification of most realistic systems was beyond the capability of such explicit-state engines. For example, a CPU with a single 32-bit register has more than $2^{32} \approx 4 \times 10^9$ possible states. Practical hardware verification via model checking had to wait for another breakthrough.

S. Chaki (✉)
Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, USA
e-mail: chaki@sei.cmu.edu

A. Gurfinkel
University of Waterloo, Waterloo, ON, Canada

© Springer International Publishing AG, part of Springer Nature 2018

E.M. Clarke et al. (eds.), *Handbook of Model Checking*,
DOI 10.1007/978-3-319-10575-8_8

The breakthrough appeared in the form of BDD-based symbolic model checking [8]. This paradigm uses a data structure called binary decision diagrams, BDDs, (see Chap. 7 and [1, 6]). BDDs are used to symbolically represent the transition relation of the automata or Kripke structures under analysis, and sets of states manipulated by the model-checking algorithm. Representing and manipulating transition relations and sets of states is sufficient for implementing model-checking algorithms for a wide range of temporal logics, for example, using the fixed-point construction described in Chap. 2.

Since its inception, BDD-based symbolic model checking has revolutionized formal verification and formal methods in profound ways. First, it has enabled practical verification of industrial systems—beginning with hardware [7] and extending to software [3]. Second, it has led to important developments for BDDs, such as new types of BDDs [22], variable-ordering heuristics [2, 24] and efficient implementations [31]. Finally, it has paved the way to other forms of symbolic model checking, especially those using efficient SAT solvers [4] and interpolants [21].

In this chapter, we present some of the key concepts and techniques in the area of BDD-based symbolic model checking. More specifically, we use reduced ordered BDDs (or ROBDDs). Unless otherwise mentioned, we use BDD to mean ROBDD. The goal of this chapter is not to be a comprehensive exposition of this rich and well-studied research area. Instead, we wish to present the basic ideas and algorithms to help someone unfamiliar with this topic get started, and to cite resources for the interested reader to find out more.

The rest of this chapter is organized as follows. Section 8.2 presents preliminary definitions borrowed from other chapters in the book. Section 8.3 presents basic concepts used in the rest of the chapter. Section 8.4 represents symbolic model checking of Kripke structures for CTL, fair CTL, and LTL. Section 8.5 presents symbolic model checking of reachability properties of push-down systems represented as Boolean programs. Section 8.6 concludes the chapter.

8.2 Preliminaries

Binary decision diagrams (BDDs) and their related concepts such as variable ordering and operations are presented in detail in Chap. 7. Therefore, we only give a brief overview of the BDD concepts and notation that are used in the rest of this chapter. Throughout, we assume that BDDs are ordered with respect to a fixed variable ordering and are reduced.

For set X, we write $\mathscr{P}(X)$ to mean the powerset of X. For a propositional formula f, let $Var(f)$ be the set of variables (a.k.a. atomic propositions) appearing in f. We assume that the reader is familiar with the basic concepts of temporal logic and its model-checking algorithms (see Chap. 2) and basic BDD operations (see Chap. 7). However, we use different notation and, for that reason, repeat some of the key definitions here. We refer the reader to earlier chapters for a more in-depth presentation of these topics.