Using MTBDDs for Composition and Model Checking of Real-Time Systems¹

Jürgen Ruf and Thomas Kropf
Institut für Rechnerentwurf und Fehlertoleranz (Prof. Dr.-Ing. D. Schmid), Universität Karlsruhe, Kaiserstr. 12, Geb. 20.20, D-76128 Karlsruhe, Germany
{Juergen.Ruf, Thomas.Kropf}@informatik.uni-karlsruhe.de
http://goethe.ira.uka.de/hvg/cats/raven/

Abstract. In this paper we show that multi-terminal BDDs (MTBDDs) are well suited to represent and manipulate interval based timed transition systems. For many timed verification tasks efficient MTBDD-based algorithms are presented. This comprises the composition of timed structures based on symbolic techniques, heuristics for state variable minimization, and a symbolic model checking algorithm. Experimental results show that in many cases our approach outperforms standard unit-delay approaches and corresponding timed automata models.

1 Introduction

For verifying real-time properties, e.g. in the domain of embedded systems, standard model checking is not directly applicable. There, quantized temporal properties have to be verified (“after 231 ns something will happen”) and the implementation model usually contains timing information like typical system delays etc. Moreover, for real systems typical delay times may vary e.g. due to fabrication tolerances and thus have to be represented by time intervals indicating minimal and maximal bounds.

Various efforts have been undertaken to extend the temporal logic and the proof algorithms to timed systems (i.e. systems containing quantized timing information). Two main approaches have to be distinguished here: those based on timed automata [4] and extensions of symbolic CTL model checking [5]. For both finding efficient algorithms and implementations is an active area of research as real-time model checking bears additional challenges compared to standard model checking:

- the model checking algorithms have to cope with time, i.e. with natural or real numbers which makes the application of propositional logic techniques based on ROBDD hard
- adding timing to state transition systems worsens the state space explosion problem, especially if a composition of timed transition systems is necessary and if time intervals are allowed, i.e. non-deterministically varying transition times.

This paper shows how MTBDDs can be used to get symbolic algorithms for all real time verification tasks: model representation, model composition, heuristic minimization of composed structures and model checking. The contributions of this paper are a consistent methodology for using MTBDDs to get symbolic real-time verification algorithms, and new composition and model checking algorithms based on symbolic state space representation and traversal techniques. Moreover, our experimental results do not only show that our method outperforms other approaches in many cases but also

¹ This work has been funded by a German research grant (DFG, SFB 358-C2)

give interesting empirical data how composition of timed systems affects the timing characteristics of the resulting systems.

2 State-of-the-Art

2.1 Timed Automata

A formalism to model real time systems are timed automata, developed by Alur and Dill [4]. In timed automata time is represented by clocks carrying real numbers and time passes in the states. A state transition is chosen based on clock predicates on the edges and input events. Specifications are given in TCTL, an extension of CTL. As an arbitrary number of clocks is possible, this is a very powerful approach. Different tools based on this theory have been presented like Kronos\(^1\) or UPPALL\(^2\). Composition of timed automata is easily possible leading to a new automaton which carries the sum of all the clocks of the original automata.

However, although deriving the composed structure is simple, the state explosion problem is only delayed to the point of model checking. Then in each state the product of the values of all clocks has to be considered. To solve this state explosion problem on-the-fly model checking techniques can be used. However, if specifications are to be proven correct which require the traversal of the complete reachable state space (e.g. mutual reachability of states) the efficiency gain is low. Moreover, the underlying proof algorithm is different from standard CTL fixed point computations and hence it is more difficult to find efficient representations. Thus only recently first efforts have been presented on how to use BDD like techniques for a symbolic state set representation. Therefore, in practice the resulting runtimes may be high especially for model checking composed structures even when using these symbolic techniques [6, 1].

2.2 Extensions to CTL Model Checking

A different approach extends CTL model checking to real-time. It is based on the observation that in many applications the expressive power of timed automata is not necessary. Usually these approaches attribute edges of the transition system with delay times (mostly natural numbers) and allow quantized timing parameters in the temporal operators, leading to CTL extensions like RTCTL (real-time CTL, [7]) or QCTL (quantized CTL, [8]). To retain the efficient BDD representation, delay times are represented by a binary encoding, added to the transition relation or by representing all transitions with a certain delay by a separate transition relation. This can be seen as a special case of timed automata where only one clock carrying natural numbers is allowed which is reset after each state transition. A tool based on this approach is Verus\(^3\). Recently new timed model checking algorithms based on multi-terminal BDDs (MTBDDs) have been proposed [9, 10]. The advantage of these approaches is that efficient implementation techniques of standard CTL model checking can be used.

Besides the reduced expressiveness compared to timed automata, the main deficiency of these approaches is structure composition. As there is only one global clock the approach of timed automata, where just the clocks of all subsystems are combined,

1. [1], http://www.imag.fr/VERIMAG/TEMPORISE/kronos/index-english.html
2. [2], http://www.brics.dk/Projects/UPPALL/
3. [3], http://www.cs.cmu.edu/~modelcheck/verus.html