Symbolic Robustness Analysis of Timed Automata

Conrado Daws[1,2,⋆] and Piotr Kordy[2]

[1] Department of Applied Mathematics
[2] Formal Methods Group
Faculty of Electrical Engineering, Mathematics and Computer Science,
University of Twente, The Netherlands
conrado.daws, piotr.kordy at ewi.utwente.nl

Abstract. We propose a symbolic algorithm for the analysis of the robustness of timed automata, that is the correctness of the model in presence of small drifts on the clocks or imprecision in testing guards. This problem is known to be decidable with an algorithm based on detecting strongly connected components on the region graph, which, for complexity reasons, is not effective in practice.

Our symbolic algorithm is based on the standard algorithm for symbolic reachability analysis using zones to represent symbolic states and can then be easily integrated within tools for the verification of timed automata models. It relies on the computation of the stable zone of each cycle in a timed automaton. The stable zone is the largest set of states that can reach and be reached from itself through the cycle. To compute the robust reachable set, each stable zone that intersects the set of explored states has to be added to the set of states to be explored.

1 Introduction

Timed automata [2] are an important formal model for the specification and analysis of real-time systems. They are a simple extension of automata with real-valued variables, called clocks, whose values increase at the same rate in the control locations, and can be reset to 0 when a discrete transition is taken. By adding a certain type of constraint on clocks to the locations and edges of the automaton, one can respectively specify the time a system is allowed to remain in a control location, and when a discrete transition can be taken. Many real-time systems have been modeled using timed automata and analyzed automatically with tools like UPPAAL [10] and KRONOS [6].

A fundamental form of system analysis is the verification of safety properties, which consists in checking whether any unsafe state is reachable. This kind of analysis is performed efficiently by the tools mentioned above with well known algorithms manipulating timed constraints, called *zones*, that can be represented as a square matrix of difference bounds (DBM). The reachability analysis is based on the *idealized* semantic assumption that all clocks advance with the same speed. However, in a real implementation of a system, clocks will be likely to drift and measure time only up to some precision.

Puri first addressed this concern in [12] where he considered drifting clocks and showed that timed automata models are not robust with respect to safety properties,

⋆ This work was originally carried out at the Nijmegen Institute for Computing and Information Sciences, Radboud University Nijmegen, and supported by the European Community Project IST-2001-35304 AMETIST

E. Asarin and P. Bouyer (Eds.): FORMATS 2006, LNCS 4202, pp. 143–155, 2006.

meaning that a model proven to be safe under the standard ideal semantics might not be safe even if clocks drift by an arbitrarily small amount. De Wulf et al. consider a semantics, called the almost ASAP semantics [8], capturing certain notion of clock imprecision that can be translated into a syntactical enlargement of the guards. They later showed in [7] that the implementability of a model under their semantics can be decided with Puri's algorithm for robustness analysis.

Both results rely on an enlarged semantics of timed automata with either drift or imprecision of clocks. They consider the set of states that are reachable for *any* drift or imprecision. If this set contains some unsafe state, then the model is considered not to be robust or implementable. The robust reachability set can be computed with the algorithm from [12] in both cases, which are thus equivalent. The algorithm is based on the structure of the limit cycles of a timed automaton, i.e. the cyclic trajectories in the underlying timed transition system. The algorithm considers the strongly connected components of the region graph because they contain the limit cycles of the timed automaton. It adds every strongly connected component that intersects the reachability set, and its successors, to the reachability set. However, because the size of the region graph is exponential in the number of clock variables and the largest constant in the constraints, the algorithm is not effective in practice.

We propose a symbolic algorithm for computing the enlarged reachability set of a timed automaton based on the standard algorithm for symbolic reachability analysis using zones to represent symbolic states. Our algorithm relies on the computation of a *stable zone* W_σ of every progress cycle σ in the timed automaton, defined as the maximal set of clock values that have successors and predecessors through any number of iterations of σ in the timed automaton. That is, $W_\sigma = \bigcap_{i\geq 0} \mathrm{post}^i_\sigma(\mathsf{True}) \cap \bigcap_{i\geq 0} \mathrm{pre}^i_\sigma(\mathsf{True})$. The stable zone has the property of reaching and being reached from any cycle in the region graph, and hence any states in a limit cycle. We modify the standard reachability algorithm such that whenever the stable zone of a cycle intersects the standard reachable set, the whole stable zone is added to the set of states to be explored.

Related work. The robustness analysis has been extended to more general type of properties, like Büchi and LTL in [4]. Other notions of robustness have been considered in the literature, like [9,11] which impose a restriction to the type of accepted traces, as opposed to the enlargement we consider here. A different modelling based approach to implementability can be found in [1].

The remaining of the paper is organized as follows. Section 2 recalls the basic standard definitions of timed automata. The robustness problem arising from an enlarged semantics is presented in Section 3. Our contribution is the subject of Section 4, where we define the stable zone of a cycle and study its main properties, which we then use in our symbolic algorithm for robustness analysis. Finally, Section 5 concludes our presentation with a summary of the main results and a discussion on future work.

2 Timed Automata

This section briefly recalls the definitions of timed automata, their semantics, reachability analysis, and region graph.