Theoretical Computer Science ••• (••••) •••–•••

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Finding minimum and maximum termination time of timed automata models with cyclic behaviour

Omar Al-Bataineh [a,*], Mark Reynolds [b], Tim French [b]

[a] Nanyang Technological University, Singapore
[b] University of Western Australia, Australia

ARTICLE INFO

Article history:
Received 31 October 2015
Received in revised form 26 November 2016
Accepted 19 December 2016
Available online xxxx
Communicated by D. Perrin

Keywords:
Worst-case execution time
Model checking
Timed automata

ABSTRACT

The paper presents a novel algorithm for computing worst case execution time (WCET) or maximum termination time of real-time systems using the timed automata (TA) model checking technology. The algorithm can work on any arbitrary diagonal-free TA and can handle more cases than previously existing algorithms for WCET computation, as it can handle cycles in TA and decide whether they lead to an infinite WCET. We show soundness of the proposed algorithm and study its complexity. To our knowledge, this is the first model checking algorithm that addresses comprehensively the WCET problem of systems with cyclic behaviour. In [7] Behrmann et al. provide an algorithm for computing the minimum cost/time of reaching a goal state in priced timed automata (PTA). The algorithm has been implemented in the well-known model checking tool UPPAAL to compute the minimum time for termination of an automaton. However, we show that in certain circumstances, when infinite cycles exist, the algorithm implemented in UPPAAL may not terminate, and we provide examples which UPPAAL fails to verify.

1. Introduction

In this paper, we consider the problem of computing the "worst case execution time" (WCET) in timed automata. Given a timed automaton $\mathcal{A}$ with a start location l_s and a final location l_f, this problem asks to compute an upper bound on the time needed to reach the final location l_f from the start location l_s. The problem is easy to solve in the case of acyclic TA [1], but cycles might introduce an unbounded WCET, that needs to be detected on-the-fly during the analysis. In general, WCET analysis is undecidable: it is undecidable to determine whether or not an execution of a system will eventually halt. However, for TA models one can use model-checking techniques to analyse the system and compute the WCET.

Typically, the infinite state-space of a timed transition system (e.g. TA) is converted into an equivalent finite state-space of a symbolic transition system called a zone graph [14,11]. In a zone graph, zones (i.e. sets of valuations of the timed automaton clocks) are used to denote symbolic states. The zone graph has been successfully used for the verification of safety and liveness properties of timed automata. Although the zone graph is precise enough to preserve the reachability properties in TA, it is too abstract to infer continuous time progress. At each step of the successor computation, the generated zones are extrapolated (abstracted) using a set of extrapolation operators and then canonicalized (tightened) in order to obtain a unique representation of the resulting zones. A test for inclusion of zones is then applied to check whether the

* Corresponding author.
E-mail address: omar.ibrahim@ntu.edu.sg (O. Al-Bataineh).

http://dx.doi.org/10.1016/j.tcs.2016.12.020
0304-3975/

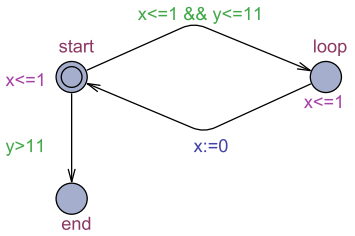

Fig. 1. $\mathcal{A}_1$: an automaton with finite cycle.

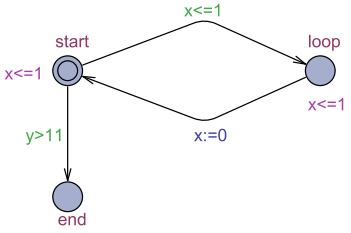

Fig. 2. $\mathcal{A}_2$: an automaton with infinite cycle.

new generated zone at a particular control location in the graph is already covered by some previously generated zones associated with that location. This helps to ensure termination of the analysis of TA even when infinite cycles exist.

However, the classical abstraction used for verification of reachability problem [10] is not correct for WCET computation, as they give abstract zones and hence result in abstract values of the execution times. To demonstrate the problem, we give in Figs. 1 and 2 two automata where both generate identical zone graphs when applying the standard zone approach for reachability analysis. The automaton $\mathcal{A}_1$ represents an automaton with finite cycle where $WCET(\mathcal{A}_1) = 12$. For this automaton, the standard zone approach can compute correctly the WCET without involving any extra check. On the other hand, the automaton $\mathcal{A}_2$ represents an automaton with an infinite cycle where $WCET(\mathcal{A}_2) = \infty$. For this automaton, the zone approach for reachability analysis fails to give the correct answer for WCET since it returns 12 instead of ∞. Note that if we disable extrapolation during the analysis, the search may not stop and we may not be able to obtain an answer.

In a previous work [1], we propose a zone-based solution to the problem of computing WCET of real-time systems modelled as TA. The proposed solution allows one to compute the WCET of TA in only one run of the zone construction instead of making repeated guesses (guided by binary search) and multiple model checking queries as done in [20]. However, in [1] we limit applicability of our solution to timed automata without infinite cycles. In the present paper, we give a more general solution to the problem that can work on any arbitrary diagonal-free TA[1] including those containing infinite cycles. Infinite cycles indeed make the computation of WCET difficult because zone extrapolation techniques are necessary to compute a finite state-space, and extrapolation prevents a straightforward computation of the WCET. The main contribution of the paper is therefore to propose an extrapolation technique that is compatible with the WCET computation. More precisely, we give the special conditions needed to define a forward zone-based reachability algorithm that terminates and computes the correct maximal time. Thus, the provided solution can be a significant break-through in computing WCET. The proposed extrapolation technique is an interesting addition to the collection of techniques for TA analysis. It is particularly useful because it improves zone extrapolation, that is one of the weak points of TA symbolic analysis.

In [7] Behrmann et al. propose an algorithm that aims to provide a solution to the minimum cost/time reachability problem in Uniformally Priced Timed Automata (UPTA) in the presence of extrapolation. However, the extrapolation step is not detailed in [7] and the implementation in UPPAAL is often unable to terminate when the model has some cycles. We give a number of examples by which we demonstrate how and why existing algorithms for computing BCET and WCET fail (including the one being now used in the tool UPPAAL). The key difficulty in developing a solution to the minimum/maximum termination time problems using the zone approach is to define an abstraction of zones that guarantees termination of the algorithm, while keeping information precise for the extra clock that is used to compute the execution time of the automaton. This involves adapting two classical operations on zones: extrapolation and canonicalization. The later was forgotten in [7] leading to non-termination.

To fix the algorithm in [7] we propose what we call the partial canonicalization technique in which the constraints involving the automaton clocks (the constraints that may be changed during extrapolation) are tightened *independently* of the constraints involving the extra clock (i.e. the constraints that are not changed during extrapolation), while the constraints involving the extra clock are tightened using the entire set of clock constraints. This guarantees both correctness and termination of the analysis.

[1] A class of TA in which the test of the form $x - y \sim c$ is disallowed, where x, y are clock variables, c is a constant, and $\sim \in \{<, \leq, =, >, \geq\}$.

Please cite this article in press as: O. Al-Bataineh et al., Finding minimum and maximum termination time of timed automata models with cyclic behaviour, Theoret. Comput. Sci. (2016), http://dx.doi.org/10.1016/j.tcs.2016.12.020