A Symbolic Decision Procedure for Robust Safety of Timed Systems

Mani Swaminathan and Martin Fränzle
Dept. of Computing Science, University of Oldenburg
{mani.swaminathan, fraenzle}@informatik.uni-oldenburg.de

Timed Automata (TA) [1] have emerged as an important formalism for the formal modelling and analysis of timed systems. Reachability analysis forms the core of TA-based verification tools such as UPPAAL [2], which implement a zone-based Forward Reachability Analysis (FRA) algorithm. This analysis, however, does not consider realistic models that are robust w.r.t imprecision such as drift in the clocks. On the other hand, existing techniques for “robust” reachability analysis of TA [3, 4, 5], which deal with such imprecisions, are not straight-forward to implement. The approaches in [3, 4] are region-based, while that of [5], though zone-based, involves a forward and backward fix-point alternation, and may not necessarily terminate.

We present here a symbolic (zone-based) FRA algorithm for deciding safety (reachability) of TA, which is robust w.r.t clock-drift, and which involves minimal overhead w.r.t the standard FRA algorithm of UPPAAL and is guaranteed to terminate in a number of iterations comparable to the standard case. The algorithm makes no assumption concerning the cycles of the TA and can be applied to closed targets, both unlike in [3, 4, 5]. However, the notion of robustness is weaker in the following sense: For a given TA with maximum clock-drift parameterized by ε > 0, with the corresponding reachable state-space being Reachε, the algorithms in [3, 4, 5] compute the set \( \bigcap_{i \geq 0} \text{Reach}^i \) and test it for empty intersection with the (closed) target. Our algorithm, on the other hand, decides whether, for any given number of iterations \( i \), there is \( \varepsilon_i > 0 \) such that Reach_{\varepsilon_i} has an empty intersection with the (not necessarily closed) target, where Reach_{\varepsilon_i} is the reachable state-space after \( i \) iterations of the transition relation under maximum perturbation \( \varepsilon_i \) of the clocks. Note that \( \varepsilon_i \) may depend on the number \( i \) of executed iterations, with \( \varepsilon_i \) decreasing (not necessarily strictly) with \( i \), and potentially tending to 0 as \( i \) tends to ∞. We however submit that this is a realistic notion of robustness, since it is reasonable to expect that the system actually runs only for finite life-time, thus tolerating strictly positive clock-drift by “discounting the future” [6].

Our algorithm computes the set \( \bigcup_{i \in \mathbb{N}} \bigcap_{\varepsilon_1 \geq 0} \text{Reach}^i_{\varepsilon_i} \) (where \( \bigcup_{i} \bigcap_{k} \) denotes region-equivalence indexed by clock ceiling \( k \)) and is sound and complete w.r.t the notion of robustness discussed earlier. The only difference between our algorithm and the standard non-robust FRA is that the time-passage operator on zones used in the standard FRA is modified to include a “neighbourhood”, which incorporates the effect of infinitesimal clock-drift for an arbitrary but finite number of iterations. The computation of the neighbourhood operator \( N_{\varepsilon}(Z) \) of a zone \( Z \) in DBM form consists of relaxing all non-strict lower bounds on clock-differences defining the zone by 1, to the next higher strict lower bounds, followed by standard \( k \)-normalization. Thus, computation of the neighbourhood of a zone incurs negligible extra cost in the FRA, being \( O(n) \), where \( n \) is the number of clocks in the TA. Termination of our procedure is guaranteed by the use of \( k \)-normalization while computing a zone’s neighbourhood.

References