Abstract:

This deliverable reports on several approaches developed inside the QUASIMODO consortium in order to bridge the gap between the mathematical semantics of models and the digital, imprecise behaviour of their implementations.

Keyword list: Implementability, verification.
Contents

1 Introduction 3

2 Robust analysis of timed automata 4
   2.1 Safety analysis ......................................................... 4
      2.1.1 Participants ..................................................... 4
      2.1.2 Contribution ...................................................... 4
   2.2 Quantitative model-checking ....................................... 4
      2.2.1 Participants ..................................................... 4
      2.2.2 Contribution ...................................................... 4
   2.3 Robustness analysis under finite life-time or resynchronization .. 4
      2.3.1 Participants ..................................................... 4
      2.3.2 Contribution ...................................................... 5

3 Symbolic algorithms for robust verification 6
   3.1 Participants .......................................................... 6
   3.2 Contribution .......................................................... 6

4 Probabilistic approach to robustness 7
   4.1 Participants .......................................................... 7
   4.2 Contribution .......................................................... 7

Bibliography 8
1 Introduction

Timed automata are governed by an idealized semantics that assumes a perfectly precise behavior of the clocks. The traditional semantics is not robust because the slightest perturbation in the timing of actions may lead to completely different behaviors of the automaton. Several recent works have considered a relaxation of this semantics, in which guards on transitions are widened by $\Delta > 0$ and clocks can drift by $\varepsilon > 0$. The relaxed semantics encompasses the imprecisions that are inevitably present in an implementation of a timed automaton, due to the finite precision of digital clocks.
2 Robust analysis of timed automata

2.1 Safety analysis

2.1.1 Participants

- Martin De Wulf and Jean-François Raskin, CFV, Université Libre de Bruxelles, Belgium
- Laurent Doyen, EPFL Lausanne, Switzerland
- Nicolas Markey, CNRS/LSV, Cachan, France

2.1.2 Contribution

In this paper, we tackle the basic problem of reachability under the relaxed semantics. We solve the safety verification problem for this robust semantics: given a timed automaton and a set of bad states, our algorithm decides if there exist positive values for the parameters \( \Delta \) and \( \varepsilon \) such that the timed automaton never enters the bad states under the relaxed semantics. We also prove that our algorithm requires polynomial space, i.e., it has the same theoretical complexity as safety verification algorithms in the classical semantics. This has been published in [4].

2.2 Quantitative model-checking

2.2.1 Participants

- Patricia Bouyer and Nicolas Marley, CNRS/LSV, Cachan, France
- Pierre-Alain Reynier, LIF, Marseille, France

2.2.2 Contribution

We have extended this result by adapting a recent (classical) model-checking algorithm for a very expressive timed temporal logic to the case of robust model checking. This algorithm is based on a translation to channel automata, that is, automata equipped with a FIFO channel with two extra operations: renaming and occurrence testing. This has been published in [3].

2.3 Robustness analysis under finite life-time or resynchronization

2.3.1 Participants

- Mani Swaminathan and Martin Fränzle, Uni. Oldenburg, Germany
- Joost-Pieter Katoen, RWTH, Aachen, Germany
2.3.2 Contribution

The unsafe states that become reachable with drifting clocks (but which are unreachable with perfect clocks) are obtained by iterating unboundedly many times through the (progress) cycles of the TA, assuming an infinite system’s life-time. Moreover, unbounded relative drift between clocks is considered which does not take into account the regular resynchronization of clocks that is performed in many implementations of real-time systems.

We address these two issues, with two main contributions:

1. Under closed guards, invariants, and targets, the standard zone-based FRA of TA performed by tools such as UPPAAL is shown to be exact for robust safety for TA with an arbitrary, but finite life-time. That is, for any $i$, there is $\varepsilon_i > 0$ such that $\text{Reach}^{\varepsilon_i}_i \cdot G = \emptyset$ where $\text{Reach}^{\varepsilon_i}_i$ is the reachable state space after $i$ iterations under maximum perturbation $\varepsilon_i$ of the clocks. Robust safety thus does not imply the existence of a homogeneous $\varepsilon > 0$ that is independent of the number of iterations, but avoids the target state $G$ by some strictly positive value of the perturbation for any arbitrary, but finite number of iterations.

2. We consider clock-drifts with the possibility if regular clock resynchronization. This results in a bounded relative clock-drift. Under the assumption of closed guards, invariants, and targets, we show that the standard zone-based FRA of TA (like in UPPAAL) is exact for robust safety of TA with regular clock resynchronization. In this case, a certification of robust safety imposes no restriction on the life-time of the system—it implies avoidance of the (closed) target by all $0 < \varepsilon < 1$ (where the $\varepsilon$ now parameterizes the maximum relative bounded clock-drift subject to periodic resynchronization) independent of the number of iterations.

This work is published as [6].
3 Symbolic algorithms for robust verification

3.1 Participants

- Alexandre David and Kim. G. Larsen, CISS, Aalborg University, Denmark
- Piotr Kordy, Rom Langerak and Jan Willem Polderman, Twente University, The Netherlands

3.2 Contribution

The algorithm for calculating the extended reachable sets under the relaxed semantics is based on detecting strongly connected components in the region automaton corresponding to a timed automaton. Whereas this shows decidability and provides a good theoretical analysis of the problem, this approach is not very practical because the region automaton is usually too large to be analysed by tools.

Therefore we have worked on a symbolic algorithm based on the notion of stable zone: a stable zone is a zone that has for arbitrarily many iterations of a certain cycle in the timed automaton both successors and predecessors in that same zone. Stable zones can be detected on-the-fly and can be calculated using standard operations on zones. Based on the analysis of stable zones an algorithm has been presented; this zone-based analysis is amenable to practical implementation.

Moreover, the original approach had the restriction that clocks are assumed to be bounded, and that each cycle in the resulting timed automaton should be a progress cycle, i.e., all clocks are reset in this cycle. Especially the latter assumption is very restrictive when considering networks of synchronizing automata. In our approach both restrictions have been removed.

The algorithm, an adaptation of the standard on-the-fly reachability algorithm, has been implemented and integrated in the state of the art tool UPPAAL. This work is under submission [5].
4 Probabilistic approach to robustness

4.1 Participants

- Christel Baier and Marcus Größer, Uni. Dresden, Germany
- Nathalie Bertrand, IRISA, Rennes, France
- Patricia Bouyer and Nicolas Markey, CNRS/LSV, Cachan, France
- Thomas Brihaye, CFV, Univ. Mons-Hainaut, Belgium

4.2 Contribution

The mathematical aspect of real-time model-checking has another disadvantage: it detects every single failure, even if it is highly unlikely to occur. While this exhaustivity is often seen as a strength of this method, it might be desirable to sometimes ignore those unlikely paths. To cope with this problem, we have defined a probabilistic semantics for timed automata. Roughly, in a given configuration, a transition that is firable only at a finite number of single dates will have zero probability if some other transition, from the same configuration, is allowed on (at least) a non-empty interval of dates.

We proposed an algorithm for almost-surely model-checking $\omega$-regular properties; we also proposed a method to compute (or approximate) the probability of an $\omega$-regular property under that semantics. These two results have been published in [1, 2].
Bibliography


