Computing maximally-permissive strategies in acyclic timed automata

Emily Clement[1,2]

[1] IRISA – Inria, CNRS & Univ. Rennes 1 – France
[2] Mitsubishi Electric R&D Centre Europe – Rennes, France

Abstract. Timed automata are a convenient mathematical model for modelling and reasoning about real-time systems. While they provide a powerful way of representing timing aspects of such systems, timed automata assume arbitrary precision and zero-delay actions; in particular, a state might be declared reachable in a timed automaton, but impossible to reach in the physical system it models. In this paper, we describe our approach to quantify and compute this precision for timed games.

1 Introduction

Timed automata [AD94] are a powerful formalism for modelling and reasoning about real-time computer systems: they offer a convenient way of modelling timing conditions (not relying on discretization) while allowing for efficient verification algorithms; as a consequence, they have been widely studied by the formal-verification community, and have been applied to numerous industrial case studies thanks to advanced tools such as Uppaal [BDL+06], TChecker [HPT19].

One drawback of timed automata is that they are a mathematical model, assuming infinite precision in the measure of time; this does not correspond to physical devices such as computers. As a consequence, properties that are proven to hold on the model may fail to hold on any implementation. As a very simple example, consider two (or even infinitely-many) consecutive actions that have to be performed at the exact same time: while this would be possible in a mathematical model, this would not be possible on a physical device.

Several approaches have attempted to address such problems, depending on the property to be checked. When considering safety properties, timing imprecisions may add new behaviours, which have to be taken into account in the safety check. In that setting, *guard enlargement* [Pur00, DDMR04] has been proposed as a way to model the fact that some timing conditions might be considered true even if they are (slightly) violated: the existence of an enlargement value for which the set of executions is safe is decidable. When dealing with reachability properties, timing imprecisions may prevent a run to be valid. A topological approach has been proposed, where a state is declared reachable only if there is a *tube of trajectories* reaching the target state [GHJ97]. Game-based approaches have also been proposed, where a state is said reachable if there is a strategy to reach this state when an opponent player is allowed to modify (up to a certain point) the values of the delays [BMS15, BFM15].

In our work, we build on the approach of [BFM15]: in that paper, the authors aim at computing *maximally-permissive* strategies for reaching a target state. While in classical timed automata, reachability is witnessed by a sequence of delays and transitions leading to a target state, here the aim is to propose *intervals* of delays, leaving it to an opponent player to decide which delay will indeed take place. Of course, the strategy has to be able to respond to any choice of the opponent, always eventually reaching the target state. Its algorithm to compute its strategy could only be achieved in the case of one-clock timed automata.

Our goal is to extend it to multiple-clocks (acyclic, or arbitrary automata) and other type of permissiveness function. We consider two type of permissiveness that we will introduce in this paper.

This paper is organized as follows: in Section 2 we introduce the type of permissiveness we consider and in Section 3 our current contribution and future work.

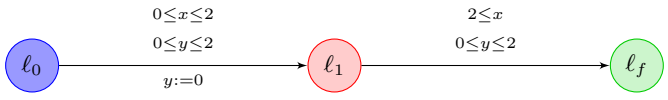
2 Our permissiveness functions

Our permissiveness function is quantified by the size of the smallest interval proposed by a strategy. We first consider the worst case, we build two players. One can proposes an interval (player) and one can propose a delay (opponent). The first one's goal is to maximize and the second one is to minimize it.

In future work, we will consider a probabilistic case,with a player and a probabilistic distribution for the opponent. The difference is that the opponent chooses the delay with a probabilistic distribution. The two approaches are the main issues of our work.

3 Our current contribution and future work

Our goal is to describe a winning strategy of the player that maximizes the permissiveness function and the strategy of the opponent that minimizes it. We compute the permissiveness function with an algorithm. Our current contribution is to describe an algorithm for the worst case in acyclic timed automaton and games. This algorithm computes the permissiveness function in non-elementary time for acyclic timed automata and in double-exponential time for linear timed automata, that are timed automata where there are only one transition available per location. To construct this algorithm, we use game theory to model the permissiveness in a timed automata and we optimize the strategies of each player to compute the permissiveness. To illustrate the notion of permissiveness, here is an example a run:



In configuration $(\ell_0, 0)$ (*i.e* we are in location ℓ_0 and the clock values 0) in figure 3, the player proposes the interval $[0, 2]$ for the unique transition available, the opponent chooses the delay $\delta = 0$. We are now in the configuration $(\ell_1, 0)$. The only interval the player can propose to reach the goal is $\{2\}$, so the opponent propose the delay $\delta = 2$. The smallest interval proposed in this run is $\{2\}$, which size is 0. The permissiveness of this run is 0. The value of the permissiveness then depends on the strategy of the player and the opponent. Our method consists in finding the best strategy for the player and the opponent.

Our future work would be to tackle to issue of cycles. We were unable to prove our intuition that there is no reason for the player to iterate a cycle. Then, another direction would be to consider the probabilistic case.

References

[AD94] Rajeev Alur and David L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126(2):183–235, April 1994.

[BDL+06] Gerd Behrmann, Alexandre David, Kim Guldstrand Larsen, John Håkansson, Paul Pettersson, Wang Yi, and Martijn Hendriks. Uppaal 4.0. In *Proceedings of the 3rd International Conference on Quantitative Evaluation of Systems (QEST'06)*, pages 125–126. IEEE Comp. Soc. Press, September 2006.

[BFM15] Patricia Bouyer, Erwin Fang, and Nicolas Markey. Permissive strategies in timed automata and games. In Gudmund Grov and Andrew Ireland, editors, *Proceedings of the 15th International Workshop on Automated Verification of Critical Systems (AVOCS'15)*, volume 72 of *Electronic Communications of the EASST*. European Association of Software Science and Technology, September 2015.

[BMS15] Patricia Bouyer, Nicolas Markey, and Ocan Sankur. Robust reachability in timed automata and games: A game-based approach. *Theoretical Computer Science*, 563:43–74, January 2015.

[DDMR04] Martin De Wulf, Laurent Doyen, Nicolas Markey, and Jean-François Raskin. Robustness and implementability of timed automata. In Yassine Lakhnech and Sergio Yovine, editors, *Proceedings of the Joint International Conferences on Formal Modelling and Analysis of Timed Systems (FORMATS'04) and Formal Techniques in Real-Time and Fault-Tolerant Systems (FTRTFT'04)*, volume 3253 of *Lecture Notes in Computer Science*, pages 118–133. Springer-Verlag, September 2004.

[GHJ97] Vineet Gupta, Thomas A. Henzinger, and Radha Jagadeesan. Robust timed automata. In Oded Maler, editor, *Proceedings of the 1997 International Workshop on Hybrid and Real-Time Systems (HART'97)*, volume 1201 of *Lecture Notes in Computer Science*, pages 331–345. Springer-Verlag, March 1997.

[HPT19] Frédéric Herbreteau, Gérald Point, and Thanh-Tung Tran. Tchecker, an open-source model-checker for timed systems, 2019.

[Pur00] Anuj Puri. Dynamical properties of timed systems. *Discrete Event Dynamic Systems*, 10(1-2):87–113, January 2000.