Research statement: What is my research about?

Emily Clement, PhD student at INRIA Rennes

October 6, 2021

1 Context & Goals

Timed automata (see fig 1) are a convenient mathematical model for modelling and reasoning about real-time systems (plane, trains, scheduling, etc). 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. Indeed in the real-world, many elements can cause imprecisions. My goal during my phD was to model these perturbations with a new semantics, called *permissive semantics*, quantify it and propose algorithm to provide maximally-permissive strategies for reachability. In my model, perturbations affect the delays of the clocks.

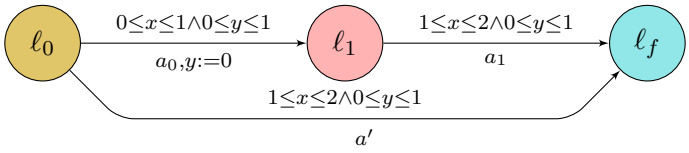

Figure 1: An acyclic timed automaton

2 Presentation of the model

In the **classical semantics**, the goal of reachability is to start from a starting location (for instance ℓ_0) and a start valuation (for instance $(0, 0)$) and reach the goal ℓ_f while proposing a pair of delay-action. For instance $(0.5, a_0)$ and then $(0.5, a_1)$. For each delay δ proposed, the clocks are increased with the same value, and the action determined with transition is taken. The increased clocks must satisfy the guard of the transition associated with the action.

The **permissive semantics** is model with a **turn-based game**. Two players are involved, a player and an opponent. The player proposes an *interval of delays* I and an action a to indicate which transition is chosen. The opponent proposes a delay δ that belongs to the interval I. Then we apply the classic semantic with the pair of delay-action (δ, a). The interval of delays represents the **perturbations on the delays**.

The size of the smallest interval that the player **proposed** during a run is called the *permissiveness* of a run.

Let us give a small example for a better understanding:

- In the classic semantics, the goal is reachable from the configuration $(\ell_0, (0,0))$: For instance by proposing first the delay 1 and the action a_0. The configuration $(\ell_1, (1,0))$ is then reached, because the clock y is reset. We then can propose the delay 1 and the action a_1. The goal is then reached and the clock valuation is $(2,1)$.
- In the permissive semantics, a pair of interval-action (called *moves*) is proposed instead of a pair of delay and action. From the same starting configuration, the move $([0,1], a_0)$ can be proposed. Whatever delay $\delta \in [0,1]$ the opponent chooses (for instance $\delta = 1$), the configuration reached is $(\ell_1, (\delta, 0))$ and the greatest interval the player will be able to propose is $[1-\delta, \min(2-\delta, 1)]$, in order to respect the guard of the second transition, whatever the delay the opponent chooses after. Here the permissiveness of the run is $\min(1, \min(2-\delta, 1) - (1-\delta))$.

The size of the smallest interval proposed during a play depends on the strategy of **both** players. The player's goal is to **maximise** it and the opponent's goal is to **minimise** it. The goal of my thesis is to quantify the size of this smallest interval with recursive functions and to optimise the strategy of the player and of the opponent. The size of the smallest interval obtained with these optimal strategies is called *the permissiveness function*. An example of its value is represented in figure 2.

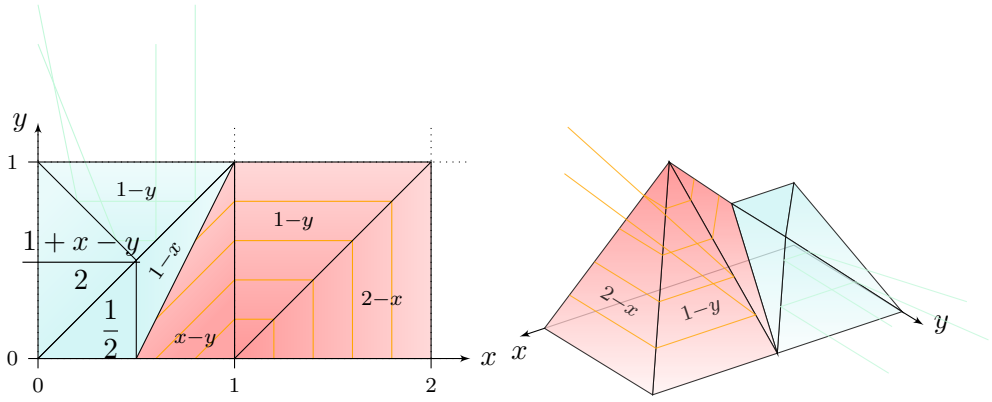

Figure 2: The permissiveness on ℓ_0 of the timed automaton described in figure 1

The main contributions are detailed in the following sections. The first ones concern **symbolic** approaches and the last one is a **numerical** approach.

3 Symbolic computation

The problems addressed was:

- **PB1**: Given an acyclic timed automaton $\mathcal{A}$ and a starting location ℓ, compute the permissiveness from (ℓ, v) for every valuation v.
- **PB2**: Given a timed automaton $\mathcal{A}$ and a starting location ℓ and a threshold $p \geq 0$, compute the set of valuations v such that the permissiveness from (ℓ, v) is greater than p ?
- **PB3**: Given a timed automaton $\mathcal{A}$ and a starting location ℓ and a threshold $p \geq 0$, compute the permissiveness from (ℓ, v) for every valuation v with precision p

3.1 Contribution 1: a symbolic algorithm for the first problem

PB1 was proven to be decidable in non-elementary time in [Cle+20], and in double-exponential time for *linear* timed automata[1]. This algorithm uses a backward algorithm that computes the permissiveness from the successors and optimise the player's strategy for each location. The complexity of this algorithm increases with the number of pieces of the piecewise affine function obtained. This algorithm reduces **PB1** to a backward exploration of a graph and a symbolic optimisation problem under polyhedron constraints. Other important contributions were to study the **form** of the permissiveness, which is:

- a piece-wise affine function in the dimension of the number of clocks of the timed automaton. The pieces are represented with **convex polyhedra** and form a **partition**.
- a 2-Lipschitz function.
- a concave function, for linear timed automata..

3.2 Contribution 2: a symbolic implementation for this previous algorithm

With the help of **PPLPY** on Python, I built a implementation of this algorithm. The main issues were the limits of convex polyhedra (non-stable with the union, etc). It is currently not published and will be available when the open-source licence will be authorised.

The goal of this implementation is to:

- give a proof of concept.
- extend to polyhedral guards.
- study the practical complexity, to see if the number of pieces of the piecewise affine function obtained exploded or not.
- study non-acyclic timed automaton. We did not prove that this algorithm finishes when the timed automaton contains cycles.

3.3 Current work: a symbolic algorithm for the second problem

These results may help us to finish my current work, which is give an efficient algorithm for **PB2**. A first naive algorithm give us a quite high complexity for linear timed automata and my goal would be to interpret this problem as a geometry problem in order to find a more efficient algorithm.

3.4 Future works

My future works after my PhD for this research subject would be:

- Find more efficient algorithms for **PB1**.
- Extend my symbolic implementation for more general model of timed automata (hybrid, etc), in order to find more general results.

[1] *i.e.* timed automata where at most one transition is available for each location

- Study problem **PB2** for acyclic timed automaton.
- Study problem **PB3** with a topological point of view: merging pieces when their volumes is small enough, for instance, would be a first way to solve this problem. As the permissiveness is a continuous function this would guarantee to stay in a controlled precision.

4 Numerical computation

The problem addressed here was an approximation problem of **PB1**:

- **PB4**: Given an acyclic timed automaton $\mathcal{A}$, a threshold $p \geq 0$ and a starting configuration (ℓ, v) with numeric value of v, compute the permissiveness from (ℓ, v), with precision p.

4.1 Contribution 3: a numerical algorithm and implementation

During my PhD, I give and implemented in Python a backtrack-based algorithm for this approximated numerical problem. Its complexity is high and **future works** would be to give some **optimisations** to reduce the runtime of this algorithm.

This algorithm samples the possible intervals and delays with a fixed step. It used a backtrack-based algorithm to store the run in a trace and optimise an approximate optimal trace. Current optimisations are, for instance, to avoid considering an interval that is smaller than the best current trace, or to sort the sampling.

This implementation is available on the gitlab of MERCE. The link is on me resume or here: `github.com/merce-fra/ECL-pyrobustness`.

References

[Cle+20] Emily Clement, Thierry Jéron, Nicolas Markey, and David Mentré. "Computing Maximally-Permissive Strategies in Acyclic Timed Automata". In: *Formal Modeling and Analysis of Timed Systems - 18th International Conference, FORMATS 2020, Vienna, Austria, September 1-3, 2020, Proceedings*. Ed. by Nathalie Bertrand and Nils Jansen. Vol. 12288. Lecture Notes in Computer Science. Springer, 2020, pp. 111–126. DOI: 10.1007/978-3-030-57628-8_7. URL: https://doi.org/10.1007/978-3-030-57628-8%5C_7.