Emptiness and Universality Problems in Timed Automata with Positive Frequency[⋆]

Nathalie Bertrand[1], Patricia Bouyer[2], Thomas Brihaye[3], and Amélie Stainer[1]

[1] INRIA Rennes, France
[2] LSV - CNRS & ENS Cachan, France
[3] Université de Mons, Belgium

Abstract. The languages of infinite timed words accepted by timed automata are traditionally defined using Büchi-like conditions. These acceptance conditions focus on the set of locations visited infinitely often along a run, but completely ignore quantitative timing aspects. In this paper we propose a natural quantitative semantics for timed automata based on the so-called frequency, which measures the proportion of time spent in the accepting locations. We study various properties of timed languages accepted with positive frequency, and in particular the emptiness and universality problems.

1 Introduction

The model of timed automata, introduced by Alur and Dill in the 90's [?] is commonly used to represent real-time systems. Timed automata consist of an extension of finite automata with continuous variables, called clocks, that evolve synchronously with time, and can be tested and reset along an execution. Despite their uncountable state space, checking reachability, and more generally ω-regular properties, is decidable *via* the construction of a finite abstraction, the so-called region automaton. This fundamental result made timed automata very popular in the formal methods community, and lots of work has been done towards their verification, including the development of dedicated tools like Kronos or Uppaal.

More recently a huge effort has been made for modelling quantitative aspects encompassing timing constraints, such as costs [?,?] or probabilities [?,?]. It is now possible to express and check properties such as: "the minimal cost to reach a given state is smaller than 3", or "the probability to visit infinitely often a given location is greater than 1/2". As a consequence, from qualitative verification, the emphasis is now put on quantitative verification of timed automata.

In this paper we propose a quantitative semantics for timed automata based on the proportion of time spent in critical states (called the *frequency*). Contrary

[⋆] This work has been partly supported by the ESF project GASICS, the ANR project ANR-2010-BLAN-0317, the Tournesol Hubert Curien partnership STP, the ARC project AUWB-2010–10/15-UMONS-3, the FRFC project 2.4515.11 and a grant from the National Bank of Belgium.

to probabilities or volume [?] that give a value to sets of behaviours of a timed automaton (or a subset thereof), the frequency assigns a real value (in $[0, 1]$) to each execution of the system. It can thus be used in a language-theoretic approach to define quantitative languages associated with a timed automaton, or boolean languages based on quantitative criteria *e.g.*, one can consider the set of timed words for which there is an execution of frequency greater than a threshold λ.

Similar notions were studied in the context of untimed systems. For finite automata, mean-payoff conditions have been investigated [?,?,?]: with each run is associated the limit average of weights encountered along the execution. Our notion of frequency extends mean-payoff conditions to timed systems by assigning to an execution the limit average of time spent in some distinguished locations. It can also be seen as a timed version of the asymptotic frequency considered in quantitative fairness games [?]. Concerning probabilistic models, a similar notion was introduced in constrained probabilistic Büchi automata yielding the decidability of the emptiness problem under the probable semantics [?]. Last, the work closest to ours deals with double-priced timed automata [?], where the aim is to synthesize schedulers which optimize on-the-long-term the reward of a system.

Adding other quantitative aspects to timed automata comes often with a cost (in terms of decidability and complexity), and it is often required to restrict the timing behaviours of the system to get some computability results, see for instance [?]. The tradeoff is then to restrict to single-clock timed automata. Beyond introducing the concept of frequency, which we believe very natural, the main contributions of this paper are the following. First of all, using a refinement of the region graph, we show how to compute the infimum and supremum values of frequencies in a given single-clock timed automaton, as well as a way to decide whether these bounds are realizable (*i.e.*, whether they are minimum and maximum respectively). The computation of these bounds together with their realizability can be used to decide the emptiness problem for languages defined by a threshold on the frequency. Moreover, in the restricted case of deterministic timed automata, it allows to decide the universality problem for these languages. Last but not least we discuss the universality problem for frequency-languages. Even under our restriction to one-clock timed automata, this problem is non-primitive recursive, and we provide a decision algorithm in the case of Zeno words when the threshold is 0. Our restriction to single-clock timed automata is crucial since at several points the techniques employed do not extend to two clocks or more. In particular, the universality problem becomes undecidable for timed automata with several clocks.

2 Definitions and preliminaries

In this section, we recall the model of timed automata, introduce the concept of frequency, and show how those can be used to define timed languages. We then compare our semantics to the standard semantics based on Büchi acceptance.

2.1 Timed automata and frequencies

We start with notations and useful definitions concerning timed automata [**?**].

Given X a finite set of clocks, a *(clock) valuation* is a mapping $v : X \to \mathbb{R}_+$. We write $\mathbb{R}_+^X$ for the set of valuations. We note $\overline{0}$ the valuation that assigns 0 to all clocks. If v is a valuation over X and $t \in \mathbb{R}_+$, then $v+t$ denotes the valuation which assigns to every clock $x \in X$ the value $v(x) + t$. For $X' \subseteq X$ we write $v_{[X' \leftarrow 0]}$ for the valuation equal to v on $X \setminus X'$ and to $\overline{0}$ on X'.

A *guard* over X is a finite conjunction of constraints of the form $x \sim c$ where $x \in X,\ c \in \mathbb{N}$ and $\sim \in \{<, \leq, =, \geq, >\}$. We denote by $G(X)$ the set of guards over X. Given g a guard and v a valuation, we write $v \models g$ if v satisfies g (defined in a natural way).

Definition 1. *A* timed automaton *is a tuple $\mathcal{A} = (L, L_0, F, \Sigma, X, E)$ such that: L is a finite set of locations, $L_0 \subseteq L$ is the set of initial locations, $F \subseteq L$ is the set of accepting locations, Σ is a finite alphabet, X is a finite set of clocks and $E \subseteq L \times G(X) \times \Sigma \times 2^X \times L$ is a finite set of edges.*

The semantics of a timed automaton $\mathcal{A}$ is given as a timed transition system $\mathcal{T}_\mathcal{A} = (S, S_0, S_F, (\mathbb{R}_+ \times \Sigma), \to)$ with set of states $S = L \times \mathbb{R}_+^X$, initial states $S_0 = \{(\ell_0, \overline{0}) \mid \ell_0 \in L_0\}$, final states $S_F = F \times \mathbb{R}_+^X$ and transition relation $\to \subseteq S \times (\mathbb{R}_+ \times \Sigma) \times S$, composed of moves of the form $(\ell, v) \xrightarrow{\tau, a} (\ell', v')$ with $\tau > 0$ whenever there exists an edge $(\ell, g, a, X', \ell') \in E$ such that $v + \tau \models g$ and $v' = (v + \tau)_{[X' \leftarrow 0]}$.

A *run* ϱ of $\mathcal{A}$ is an infinite sequence of moves starting in some $s_0 \in S_0$, *i.e.*, $\varrho = s_0 \xrightarrow{\tau_0, a_0} s_1 \cdots \xrightarrow{\tau_k, a_k} s_{k+1} \cdots$. A *timed word* over Σ is an element $(t_i, a_i)_{i \in \mathbb{N}}$ of $(\mathbb{R}_+ \times \Sigma)^\omega$ such that $(t_i)_{i \in \mathbb{N}}$ is increasing. The timed word is said to be *Zeno* if the sequence $(t_i)_{i \in \mathbb{N}}$ is bounded from above. The timed word associated with ϱ is $w = (t_0, a_0) \dots (t_k, a_k) \dots$ where $t_i = \sum_{j=0}^{i} \tau_j$ for every i. A timed automaton $\mathcal{A}$ is *deterministic* whenever, given two edges $(\ell, g_1, a, X'_1, \ell')$ and $(\ell, g_2, a, X'_2, \ell')$ in E, $g_1 \wedge g_2$ cannot be satisfied. In this case, for every timed word w, there is at most one run reading w. An example of a (deterministic) timed automaton is given in Fig. **??**. As a convention locations in F will be depicted in black.

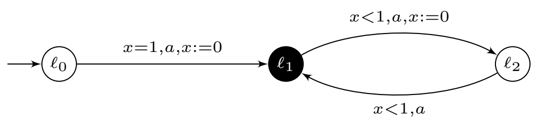


Fig. 1. Example of a timed automaton $\mathcal{A}$ with $L_0 = \{\ell_0\}$ and $F = \{\ell_1\}$.

Definition 2. *Given $\mathcal{A} = (L, L_0, F, \Sigma, X, E)$ a timed automaton and a run $\varrho = (\ell_0, v_0) \xrightarrow{\tau_0, a_0} (\ell_1, v_1) \xrightarrow{\tau_1, a_1} (\ell_2, v_2) \cdots$ of $\mathcal{A}$, the* frequency *of F along ϱ, denoted $\mathsf{freq}_\mathcal{A}(\varrho)$, is defined as $\limsup_{n \to \infty} (\sum_{i \leq n \mid \ell_i \in F} \tau_i) / (\sum_{i \leq n} \tau_i)$.*

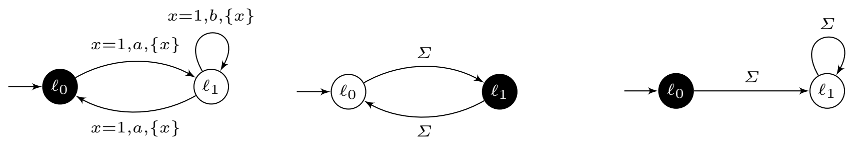
Note that the choice of lim sup is arbitrary, and the choice of lim inf would be as relevant. Furthermore notice that the limit may not exist in general.

A timed word w is said *accepted with positive frequency* by $\mathcal{A}$ if there exists a run ϱ which reads w and such that $\mathsf{freq}_{\mathcal{A}}(\varrho)$ is positive. The *positive-frequency language* of $\mathcal{A}$ is the set of timed words that are accepted with positive frequency by $\mathcal{A}$. Note that we could define more generally languages where the frequency of each word should be larger than some threshold λ, but even though some of our results apply to this more general framework we prefer focusing on languages with positive frequency.

Example 3. We illustrate the notion of frequency on runs of the deterministic timed automaton $\mathcal{A}$ of Fig. **??**. First, the only run in $\mathcal{A}$ 'reading' the word $(1, a).((\frac{1}{3}, a).(\frac{1}{3}, a))^*$ has frequency $\frac{1}{2}$ because the sequence $\frac{n/3}{1+(2n)/3}$ converges to $\frac{1}{2}$. Second, the Zeno run reading $(1, a).(((\frac{1}{2^k}, a).(\frac{1}{2^k}, a))^k)_{k\geq 1}$ in $\mathcal{A}$ has frequency $\frac{1}{3}$ since the sequence $\frac{\sum_{k\geq 1} 1/2^k}{1+\sum_{k\geq 1} 1/2^{k-1}}$ converges to $\frac{1}{3}$. Finally, the run in $\mathcal{A}$ reading the word $(1, a).(((\frac{1}{2}, a).(\frac{1}{4}, a))^{2^{2k}}.((\frac{1}{4}, a).(\frac{1}{2}, a))^{2^{2k+1}})_{k\geq 1}$ has frequency $\frac{4}{9}$. Note that the sequence under consideration does not converge, but its lim sup is $\frac{4}{9}$.

2.2 A brief comparison with usual semantics

The usual semantics for timed automata considers a Büchi acceptance condition. We naturally explore differences between this usual semantics, and the one we introduced based on positive frequency. The expressiveness of timed automata under those acceptance conditions is not comparable, as witnessed by the automaton represented in Fig. **??**: on the one hand, its positive-frequency language is not timed-regular (*i.e.* accepted by a timed automaton with a standard Büchi acceptance condition), and on the other hand, its Büchi language cannot be recognized by a timed automaton with a positive-frequency acceptance condition.



(a) Expressiveness. (b) Universality (non-Zeno). (c) Universality (Zeno).

Fig. 2. Automata for the comparison with the usual semantics.

The contribution of this paper is to study properties of the positive-frequency languages. We will show that we can get very fine information on the set of frequencies of runs in *single-clock* timed automata, which implies the decidability of the emptiness problem for positive-frequency languages. We also show that our technics do not extend to multi-clock timed automata.

We will also consider the universality problem and variants thereof (restriction to Zeno or non-Zeno timed words). On the one hand, clearly enough, a (non-Zeno)-universal timed automaton with a positive-frequency acceptance condition is (non-Zeno)-universal for the classical Büchi-acceptance. The timed automaton of Fig. **??** is a counterexample to the converse. On the other hand, a Zeno-universal timed automaton under the classical semantics is necessarily Zeno-universal under the positive-frequency acceptance condition, but the automaton depicted in Fig. **??** shows that the converse does not hold.

3 Set of frequencies of runs in one-clock timed automata

In this section, we give a precise description of the set of frequencies of runs in single-clock timed automata. To this aim, we use the *corner-point abstraction* [**?**], a refinement of the region abstraction, and exploit the links between frequencies in the timed automaton and ratios in its corner-point abstraction. We fix a single-clock timed automaton $\mathcal{A} = (L, L_0, F, \Sigma, \{x\}, E)$.

3.1 The corner-point abstraction

Even though the corner-point abstraction can be defined for general timed automata [**?**], we focus on the case of single-clock timed automata.

If M is the largest constant appearing in the guards of $\mathcal{A}$, the usual *region abstraction* of $\mathcal{A}$ is the partition $Reg_{\mathcal{A}}$ of the set of valuations $\mathbb{R}_+$ made of the singletons $\{i\}$ for $0 \leq i \leq M$, the open intervals $(i, i+1)$ with $0 \leq i \leq M-1$ and the unbounded interval (M, ∞) represented by $\perp$. A piece of this partition is called a *region*. The corner-point abstraction refines the region abstraction by associating *corner-points* with regions. The singleton regions have a single corner-point represented by $\bullet$ whereas the open intervals $(i, i+1)$ have two corner-points $\bullet\!-$ (the left end-point of the interval) and $-\!\bullet$ (the right end-point of the interval). Finally, the region $\perp$ has a single corner-point denoted $\alpha_\perp$. We write (R, α) for the region R pointed by the corner α and $(R, \alpha) + 1$ denotes its *direct time successor* defined by:

$$(R,\alpha)+1 = \begin{cases} ((i,i+1), \bullet\!-) & \text{if } (R,\alpha) = (\{i\}, \bullet) \text{ with } i < M, \\ ((i,i+1), -\!\bullet) & \text{if } (R,\alpha) = ((i,i+1), \bullet\!-), \\ (\{i+1\}, \bullet) & \text{if } (R,\alpha) = ((i,i+1), -\!\bullet), \\ (\perp, \alpha_\perp) & \text{if } (R,\alpha) = (\{M\}, \bullet) \text{ or } (\perp, \alpha_\perp). \end{cases}$$

Using these notions, we define the corner-point abstraction as follows.

Definition 4. *The* (unweighted) corner-point abstraction *of $\mathcal{A}$ is the finite automaton $\mathcal{A}_{cp} = (L_{cp}, L_{0,cp}, F_{cp}, \Sigma_{cp}, E_{cp})$ where $L_{cp} = L \times Reg_{\mathcal{A}} \times \{\bullet, \bullet\!-, -\!\bullet, \alpha_\perp\}$ is the set of states, $L_{0,cp} = L_0 \times \{0\} \times \{\bullet\}$ is the set of initial states, $F_{cp} = F \times Reg_{\mathcal{A}} \times \{\bullet, \bullet\!-, -\!\bullet, \alpha_\perp\}$ is the set of accepting states, $\Sigma_{cp} = \Sigma \cup \{\varepsilon\}$, and $E_{cp} \subseteq L_{cp} \times \Sigma_{cp} \times L_{cp}$ is the finite set of edges defined as the union of discrete transitions and idling transitions:*

– discrete transitions: $(\ell, R, \alpha) \xrightarrow{a} (\ell', R', \alpha')$ *if* α *is a corner-point of* R *and there exists a transition* $\ell \xrightarrow{g,a,X'} \ell'$ *in* $\mathcal{A}$*, such that* $R \subseteq g$ *and* $(R', \alpha') = (R, \alpha)$ *if* $X' = \emptyset$*, otherwise* $(R', \alpha') = (\{0\}, \bullet)$*,*
– idling transitions: $(\ell, R, \alpha) \xrightarrow{\varepsilon} (\ell, R', \alpha')$ *if* α *(resp.* α'*) is a corner-point of* R *(resp.* R'*) and* $(R', \alpha') = (R, \alpha) + 1$.

We decorate this finite automaton with two weights for representing frequencies, one which we call the cost, and the other which we call the reward (by analogy with double-priced timed automata in [?]). The *(weighted) corner-point abstraction* $\mathcal{A}_{cp}^F$ is obtained from $\mathcal{A}_{cp}$ by labeling idling transitions in $\mathcal{A}_{cp}$ as follows: transitions $(\ell, R, \alpha) \xrightarrow{\varepsilon} (\ell, R, \alpha')$ with $(R, \alpha') = (R, \alpha) + 1$ ($\alpha' = \alpha + 1$ for short) are assigned cost 1 (resp. cost 0) and reward 1 if $\ell \in F$ (resp. $\ell \notin F$), and all other transitions are assigned both cost and reward 0. To illustrate this definition, the corner-point abstraction of the timed automaton in Fig. **??** is represented in Fig. **??**.

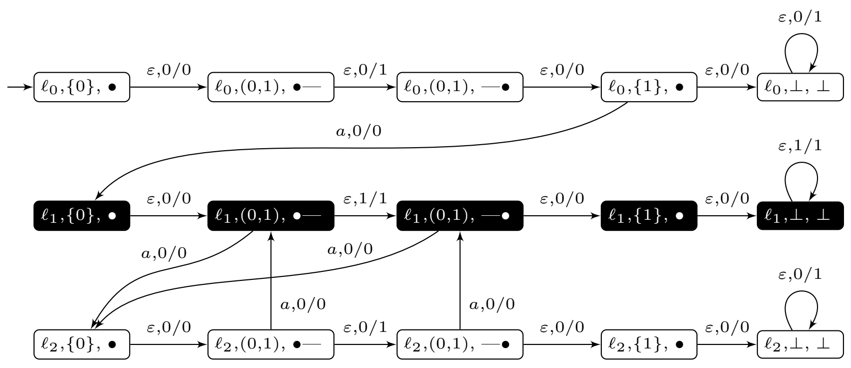


Fig. 3. The corner-point abstraction $\mathcal{A}_{cp}^F$ of $\mathcal{A}$ represented Fig. **??**.

There will be a correspondence between runs in $\mathcal{A}$ and runs in $\mathcal{A}_{cp}$. As time is increasing in $\mathcal{A}$ we forbid runs in $\mathcal{A}_{cp}$ where two actions have to be made in 0-delay (this is easy to do as there should be no sequence $\ldots \xrightarrow{\sigma} (\ell, R, \alpha) \xrightarrow{\sigma'} \ldots$, where both σ and σ' are actions and R is a punctual region).

Given π a run in $\mathcal{A}_{cp}^F$ the *ratio* of π, denoted $\mathsf{Rat}(\pi)$, is defined, provided it exists, as the lim sup of the ratio of accumulated costs divided by accumulated rewards for finite prefixes. Run π is said *reward-converging* (resp. *reward-diverging*) if the accumulated reward along π is bounded (resp. unbounded). Reward-converging runs in $\mathcal{A}_{cp}^F$ are meant to capture Zeno behaviours of $\mathcal{A}$.

Given ϱ a run in $\mathcal{A}$ we denote by $\mathsf{Proj}_{cp}(\varrho)$ the set of all runs in $\mathcal{A}_{cp}^F$ compatible with ϱ in the following sense. We assume $\varrho = (\ell_0, v_0) \xrightarrow{\tau_0, a_0} (\ell_1, v_1) \xrightarrow{\tau_1, a_1} \cdots$, where move $(\ell_i, v_i) \xrightarrow{\tau_i, a_i} (\ell_{i+1}, v_{i+1})$ comes from an edge e_i. A run[4] $\pi = (\ell_0, R_0^1, \alpha_0^1) \to (\ell_0, R_0^2, \alpha_0^2) \to \cdots \to (\ell_0, R_0^{k_0}, \alpha_0^{k_0}) \to (\ell_1, R_1^1, \alpha_1^1) \to \cdots \to (\ell_1, R_1^{k_1}, \alpha_1^{k_1}) \cdots$ of $\mathcal{A}_{cp}^F$ is in $\mathsf{Proj}_{cp}(\varrho)$ if for all indices $n \geq 0$:

- for all $i \leq k_n$, α_n^i is a corner-point of R_n^i,
- for all $i \leq k_n - 1$, $(R_n^{i+1}, \alpha_n^{i+1}) = (R_n^i, \alpha_n^i) + 1$,
- $(R_{n+1}^1, \alpha_{n+1}^1)$ is the successor pointed-region of $(R_n^{k_n}, \alpha_n^{k_n})$ by transition e_n (that is $(R_{n+1}^1, \alpha_{n+1}^1) = (\{0\}, \bullet)$ if e_n resets the clock x and otherwise $(R_{n+1}^1, \alpha_{n+1}^1) = (R_n^{k_n}, \alpha_n^{k_n})$),
- $v_n \in R_n^1$ and if $R_n^{k_n} \neq \bot$, $v_n + \tau_n \in R_n^{k_n}$,
- if $R_n^{k_n} = \bot$, the sum μ_n of the rewards since region $\{0\}$ has been visited for the last time has to be equal to $\lfloor v_n + \tau_n \rfloor$ or $\lceil v_n + \tau_n \rceil$.[5] Note that μ_n can be seen as the abstraction of the valuation v_n.

Remark 5. As defined above, the size of $\mathcal{A}_{cp}^F$ is exponential in the size of $\mathcal{A}$ because the number of regions is $2M$ (which is exponential in the binary encoding of M). We could actually take a rougher version of the regions [**?**], where only constants appearing in $\mathcal{A}$ should take part in the region partition. This partition, specific to single-clock timed automata is only polynomial in the size of $\mathcal{A}$. We choose to simplify the presentation by considering the standard unit intervals.

We will now see that the corner-point abstraction is a useful tool to deduce properties of the set of frequencies of runs in the original timed automata.

3.2 From $\mathcal{A}$ to $\mathcal{A}_{cp}^F$, and *vice-versa*

We first show that given a run ϱ of $\mathcal{A}$, there exists a run in $\mathsf{Proj}_{cp}(\varrho)$, whose ratio is smaller (resp. larger) than the frequency of ϱ.

Lemma 6 (From $\mathcal{A}$ to $\mathcal{A}_{cp}^F$). *For every run ϱ in $\mathcal{A}$, there exist π and π' in $\mathcal{A}_{cp}^F$ that can effectively be built and belong to $\mathsf{Proj}_{cp}(\varrho)$ such that:*

$$\mathsf{Rat}(\pi) \leq \mathsf{freq}_{\mathcal{A}}(\varrho) \leq \mathsf{Rat}(\pi').$$

Run π (resp. π') minimizes (resp. maximizes) the ratio among runs in $\mathsf{Proj}_{cp}(\varrho)$.

Such two runs of $\mathcal{A}_{cp}^F$ can be effectively built from ϱ, through the so-called *contraction* (resp. *dilatation*) operations. Intuitively it consists in minimizing (resp. maximizing) the time elapsed in F-locations.

Note that the notion of contraction cannot be adapted to the case of timed automata with several clocks, as illustrated by the timed automaton in Fig. **??**. Consider indeed the run alternating delays $(\frac{1}{2} + \frac{1}{n})$ and $1 - (\frac{1}{2} + \frac{1}{n})$ for $n \in \mathbb{N}$, and switching between the left-most cycle $(\ell_1 - \ell_2 - \ell_1)$ and the right-most cycle $(\ell_3 - \ell_4 - \ell_3)$ following the rules: in round k, take 2^{2k} times the cycle $\ell_1 - \ell_2 - \ell_1$, then switch to ℓ_3 and take 2^{2k+1} times the cycle $\ell_3 - \ell_4 - \ell_3$ and return back to

[4] For simplicity, we omit here the transitions labels

[5] Roughly, in the unbounded region $\bot$, the number of times an idling transition is taken should reflect how 'big' the delay τ_n is.

ℓ_1 and continue with round $k+1$. This run cannot have any contraction since its frequency is $\frac{1}{2}$, whereas all its projections in the corner-point abstraction have ratio $\frac{2}{3}$, the $\limsup$ of a non-converging sequence. This strange behavior is due to the fact that the delays in ℓ_1 and ℓ_3 need to be smaller and smaller, and this converging phenomenon requires at least two clocks.

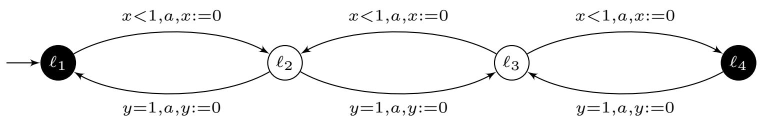


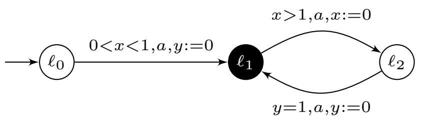
Fig. 4. A counterexample with two clocks for Lemma **??**.

We now want to know when and how runs in $\mathcal{A}_{cp}^{F}$ can be lifted to $\mathcal{A}$. To that aim we distinguish between reward-diverging and reward-converging runs.

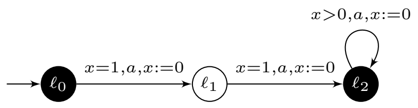
Lemma 7 (From $\mathcal{A}_{cp}^{F}$ to $\mathcal{A}$, reward-diverging case). *For every reward-diverging run π in $\mathcal{A}_{cp}^{F}$, there exists a non-Zeno run ϱ in $\mathcal{A}$ such that $\pi \in \mathsf{Proj}_{cp}(\varrho)$ and $\mathsf{freq}_{\mathcal{A}}(\varrho) = \mathsf{Rat}(\pi)$.*

Proof (Sketch). The key ingredient is that given a reward-diverging run π in $\mathcal{A}_{cp}^{F}$, for every $\varepsilon > 0$, one can build a non-Zeno run ϱ_ε of $\mathcal{A}$ with the following strong property: for all $n \in \mathbb{N}$, the valuation of the n-th state along ϱ_ϵ is $\frac{\epsilon}{2^n}$-close to the abstract valuation in the corresponding state in π. The accumulated reward along π diverges, hence $\mathsf{freq}_{\mathcal{A}}(\varrho_\varepsilon)$ is equal to $\mathsf{Rat}(\pi)$. $\square$

The restriction to single-clock timed automata is crucial in Lemma **??**. Indeed, consider the two-clocks timed automaton depicted in Fig. **??**. In its corner-point abstraction there exists a reward-diverging run π with $\mathsf{Rat}(\pi) = 0$, however every run ϱ satisfies $\mathsf{freq}_{\mathcal{A}}(\varrho) > 0$.



(a) A counterexample with two clocks.



(b) Zeno case.

Fig. 5. Counterexamples to extensions of Lemma **??**.

Lemma 8 (From $\mathcal{A}_{cp}^{F}$ to $\mathcal{A}$, reward-converging case). *For every reward-converging run π in $\mathcal{A}_{cp}^{F}$, if $\mathsf{Rat}(\pi) > 0$, then for every $\varepsilon > 0$, there exists a Zeno run ϱ_ε in $\mathcal{A}$ such that $\pi \in \mathsf{Proj}_{cp}(\varrho_\varepsilon)$ and $|\mathsf{freq}_{\mathcal{A}}(\varrho_\varepsilon) - \mathsf{Rat}(\pi)| < \varepsilon$.*

Proof (Sketch). A construction similar to the one used in the proof of Lemma **??** is performed. Note however that the result is slightly weaker, since in the reward-converging case, one cannot neglect imprecisions (even the smallest) forced *e.g.*, by the prohibition of the zero delays. □

Note that Lemma **??** does not hold in case $\mathsf{Rat}(\pi) = 0$, where we can only derive that the set of frequencies of runs ϱ such that $\pi \in \mathsf{Proj}_{cp}(\varrho)$ is either $\{0\}$ or $\{1\}$ or included in $(0, 1)$. Also an equivalent to Lemma **??** for Zeno runs (even in the single-clock case!) is hopeless. The timed automaton $\mathcal{A}$ depicted in Fig. **??**, where $F = \{\ell_0, \ell_2\}$ is a counterexample. Indeed, in $\mathcal{A}^F_{cp}$ there is a reward-converging run π with $\mathsf{Rat}(\pi) = \frac{1}{2}$, whereas all Zeno runs in $\mathcal{A}$ have frequency larger than $\frac{1}{2}$.

3.3 Set of frequencies of runs in $\mathcal{A}$

We use the strong relation between frequencies in $\mathcal{A}$ and ratios in $\mathcal{A}^F_{cp}$ proven in the previous subsection to establish key properties of the set of frequencies.

Theorem 9. *Let $\mathcal{F}_\mathcal{A} = \{\mathsf{freq}_\mathcal{A}(\varrho) \mid \varrho \text{ run of } \mathcal{A}\}$ be the set of frequencies of runs in $\mathcal{A}$. We can compute* $\inf \mathcal{F}_\mathcal{A}$ *and* $\sup \mathcal{F}_\mathcal{A}$*. Moreover we can decide whether these bounds are reached or not. Everything can be done in NLOGSPACE.*

The above theorem is based on the two following lemmas dealing respectively with the set of non-Zeno and Zeno runs in $\mathcal{A}$.

Lemma 10 (non-Zeno case). *Let $\{C_1, \cdots, C_k\}$ be the set of reachable SCCs of $\mathcal{A}^F_{cp}$. The set of frequencies of non-Zeno runs of $\mathcal{A}$ is then $\cup_{1\le i\le k}[m_i, M_i]$ where m_i (resp. M_i) is the minimal (resp. maximal) ratio for a reward-diverging cycle in C_i.*

Proof (Sketch). First, the set of ratios of reward-diverging runs in $\mathcal{A}^F_{cp}$ is exactly $\cup_{1\le i\le k}[m_i, M_i]$. Indeed, given two extremal cycles c_m and c_M of ratios m and M in an SCC C of $\mathcal{A}^F_{cp}$, we show that every ratio $m \le r \le M$ can be obtained as the ratio of a run ending in C by combining in a proper manner c_m and c_M. Then, using Lemmas **??** and **??** we derive that the set of frequencies of non-Zeno runs in $\mathcal{A}$ coincides with the set of ratios of reward-diverging runs in $\mathcal{A}^F_{cp}$. □

Lemma 11 (Zeno case). *Given π a reward-converging run in $\mathcal{A}^F_{cp}$, it is decidable whether there exists a Zeno run ϱ such that π is the contraction of ϱ and* $\mathsf{freq}_\mathcal{A}(\varrho) = \mathsf{Rat}(\pi)$.

Proof (Sketch). Observe that every fragment of π between reset transitions can be considered independently, since compensations cannot occur in Zeno runs: even the smallest deviation (such as a delay ε in $\mathcal{A}$ instead of a cost 0 in π) will introduce a difference between the ratio and the frequency. A careful inspection of cases allows one to establish the result stated in the lemma. □

Using Lemmas **??** and **??**, let us briefly explain how we derive Theorem **??**. For each SCC C of the corner-point abstraction $\mathcal{A}_{cp}^{F}$, the bounds of the set of frequencies of runs whose contraction ends up in C can be computed thanks to the above lemmas. We can also furthermore decide whether these bounds can be obtained by a real run in $\mathcal{A}$. The result for the global automaton follows.

Remark 12. The link between $\mathcal{A}$ and $\mathcal{A}_{cp}^{F}$ differs in several aspects from [**?**]. First, a result similar to Lemma **??** was proven, but the runs π and π' were not in $\mathsf{Proj}_{cp}(\varrho)$, and more importantly it heavily relied on the reward-diverging hypothesis. Then the counter-part of Theorem **??** was weaker in [**?**] as there was no way to decide whether the bounds were reachable or not.

4 Emptiness and Universality Problems

The emptiness problem. In our context, the emptiness problem asks, given a timed automaton $\mathcal{A}$ whether there is a timed word which is accepted by $\mathcal{A}$ with positive frequency. We also consider variants where we focus on non-Zeno or Zeno timed words. As a consequence of Theorem **??**, we get the following result.

Theorem 13. *The emptiness problem for infinite (resp. non-Zeno, Zeno) timed words in single-clock timed automata is decidable. It is furthermore NLOGSPACE-Complete.*

Note that the problem is open for timed automata with 2 clocks or more.

The universality problem. We now focus on the universality problem, which asks, whether all timed words are accepted with positive frequency in a given timed automaton. We also consider variants thereof which distinguish between Zeno and non-Zeno timed words. Note that these variants are incomparable: there are timed automata that, with positive frequency, recognize all Zeno timed words but not all non-Zeno timed words, and *vice-versa*.

A first obvious result concerns deterministic timed automata. One can first check syntactically whether all infinite timed words can be read (just locally check that the automaton is complete). Then we notice that considering all timed words exactly amounts to considering all runs. Thanks to Theorem **??**, one can decide, in this case, whether there is or not a run of frequency 0. If not, the automaton is universal, otherwise it is not universal.

Theorem 14. *The universality problem for infinite (resp. non-Zeno, Zeno) timed words in deterministic single-clock timed automata is decidable. It is furthermore NLOGSPACE-Complete.*

Remark 15. Note that results similar to Theorems **??** and **??** hold when considering languages defined with a threshold λ on the frequency.

If we relax the determinism assumption this becomes much harder!

Theorem 16. *The universality problem for infinite (resp. non-Zeno, Zeno) timed words in a single-clock timed automaton is non-primitive recursive. If two clocks are allowed, this problem is undecidable.*

Proof (Sketch). The proof is done by reduction to the universality problem for finite words in timed automata (which is known to be undecidable for timed automata with two clocks or more [?] and non-primitive recursive for one-clock timed automata [?]). Given a timed automaton $\mathcal{A}$ that accepts finite timed words, we construct a timed automaton $\mathcal{B}$ with an extra letter c which will be interpreted with positive frequency. From all accepting locations of $\mathcal{A}$, we allow $\mathcal{B}$ to read c and then accept everything (with positive frequency). The construction is illustrated on Fig. **??**. It is easy to check that $\mathcal{A}$ is universal over Σ iff $\mathcal{B}$ is universal over $\Sigma \cup \{c\}$. □

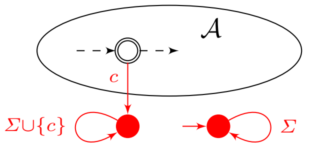


Fig. 6.

Theorem 17. *The universality problem for Zeno timed words with positive frequency in a one-clock timed automaton is decidable.*

Proof (Sketch). This decidability result is rather involved and requires some technical developments for which there is no room here. It is based on the idea that for a Zeno timed word to be accepted with positive frequency it is (necessary and) sufficient to visit an accepting location once. Furthermore the sequence of timestamps associated with a Zeno timed word is converging, and we can prove that from some point on, in the automaton, all guards will be trivially either verified or denied: for instance if the value of the clock is 1.4 after having read a prefix of the word, and if the word then converges in no more than 0.3 time units, then only the constraint $1 < x < 2$ will be satisfied while reading the suffix of the word, unless the clock is reset, in which case only the constraint $0 < x < 1$ will be satisfied. Hence the algorithm is composed of two phases: first we read the prefix of the word (and we use a now standard abstract transition system to do so, see [?]), and then for the tail of the Zeno words, the behaviour of the automaton can be reduced to that of a finite automaton (using the above argument on tails of Zeno words). □

5 Conclusion

In this paper we introduced a notion of (positive-)frequency acceptance for timed automata and studied the related emptiness and universality problems. This semantics is not comparable to the classical Büchi semantics. For deterministic single-clock timed automata, emptiness and universality are decidable by investigating the set of possible frequencies based on the corner-point abstraction. For (non-deterministic) single-clock timed automata, the universality problem restricted to Zeno timed words is decidable but non-primitive recursive. The restriction to single-clock timed automata is justified on the one hand by the undecidability of the universality problem in the general case. On the other hand, the techniques we employ to study the set of possible frequencies do not extend to timed automata with several clocks. A remaining open question is the decidability status of the universality problem for non-Zeno timed words, which is only known to be non-primitive recursive. Further investigations include a deeper study of frequencies in timed automata with multiple clocks, and also the extension of this work to languages accepted with some frequency larger than a given threshold.