Diagnosing timed automata using timed markings

Patricia Bouyer, Léo Henry, Samy Jaziri, Thierry Jéron, Nicolas Markey

Noname manuscript No.
(will be inserted by the editor)

Diagnosing timed automata using timed markings

Patricia Bouyer · Léo Henry · Samy Jaziri · Thierry Jéron · Nicolas Markey

the date of receipt and acceptance should be inserted later

Abstract We consider the problems of efficiently diagnosing (and predicting) what did (and will) happen after a given sequence of observations of the execution of a partially-observable one-clock timed automaton. This is made difficult by the facts that timed automata are infinite-state systems, and that they can in general not be determinized.

We introduce *timed markings* as a formalism to keep track of the evolution of the set of reachable configurations over time. We show how timed markings can be used to efficiently represent the closure under silent transitions of such automata. We report on our implementation of this approach compared to the approach of [Tripakis, *Fault diagnosis for timed automata*, 2002], and provide some insight to a generalization of our approach to n-clock timed automata.

1 Introduction

Formal methods in verification. Because of the wide range of applications of computer systems, and of their increasing complexity, the use of formal methods for checking their correct behaviours has become essential [20,12]. Numerous approaches have been introduced and extensively studied over the last 40 years, and mature tools now exist and are used in practice. Most of these approaches rely on building mathematical models, such as automata and extensions thereof, in order to represent and reason about the behaviours of those systems; various algorithmic techniques are then applied in order to ensure correctness of those behaviours, such as *model checking* [13,14], *deductive verification* [22,15].

Online verification. The techniques listed above mainly focus on assessing correctness of the set of all behaviours of the system in an *offline* manner. This is usually very costly in terms of computation, and sometimes too strong a requirement. *Runtime verification* instead aims at checking properties of a running system [24]. *Fault diagnosis* is a prominent problem in runtime verification: it consists in (deciding the existence and) building a diagnoser, whose role is to monitor real executions of a partially-observable system, and detect *online*, as early as possible, whether some property holds (e.g., whether some unobservable fault has occurred) [25,27]. For finite-state models, a diagnoser can usually be built by determinizing a model of the system, using the powerset construction; it will keep track of all possible states that can be reached after each (observable) step of the system, thereby computing whether a fault may or must have occurred. The related problem of *prediction*, a.k.a. prognosis, (that e.g. no faults may occur in the near future) [17], is also of particular interest in runtime verification, and can be solved using similar techniques. *Conformance testing* is another online verification technique whose aim is to check whether the behaviours of a (black-box) running system conforms to its specification. To do that a tester runs test cases which control inputs of the system and observe that outputs conform to the specification. In its model-based declination [26], the specification is formal, e.g. an automaton, and since the system and specification are partially observable, the generation of tests (online or offline) may also rely on determinization.

Work supported by ERC project EQualIS and ANR project TickTac.

P. Bouyer · S. Jaziri
LSV – CNRS & Univ. Paris-Saclay – France

L. Henry · T. Jéron · N. Markey
IRISA – Univ Rennes & CNRS & INRIA – France

Verifying real-time systems. Real-time constraints often play an important role for modelling and specifying correctness of computer systems. Discrete models, such as finite-state automata, are not adequate to model such real-time constraints; timed automata [1], developed at the end of the 1980's, provide a convenient framework for both representing and efficiently reasoning about computer systems subject to real-time constraints. Efficient offline verification techniques for timed automata have been developed and implemented [4, 3]. Diagnosis of timed automata however has received less attention; this problem is made difficult by the fact that timed automata can in general not be determinized [28,16]. This has been circumvented by either restricting to classes of determinizable timed automata [9], or by transforming the detection problem into a *state estimation problem* and keeping track of all possible configurations of the automaton after a (finite) execution [27]. The latter approach is computationally very expensive, as one step consists in maintaining the set of all configurations that can be reached by following (arbitrarily long) sequences of unobservable transitions; this is achieved by a classical zone-based algorithm for computing reachable configurations. Moreover, the set of possible configurations is updated only when a new event is observed (or after a timeout), which may significantly delay the detection of a fault. This limits the applicability of the approach. The situation is similar for model-based conformance testing for timed automata, where test generation is either online, relying on state estimation techniques similar to [27], or is offline and restricts to determinizable classes, except [23,6] which both rely on approximate determinization.

Clearly, the state estimation problem is central to several runtime verification techniques as soon as partial observation is considered. For timed automata models, it requires to update the set of configurations, either after an observable action, or a sequence of silent actions occurring in a given delay (also called τ-closure). Efficiently solving this problem is the main challenge addressed in this paper.

Our contribution. In this paper, we develop a novel technique for efficiently computing, at runtime, the set of all possible configurations in which a partially-observable one-clock timed automaton can be.

The main ingredient of our approach is the notion of *linear timed sets*: intuitively, a timed set is a set that evolves over time (formally, it is a mapping $f\colon d \in \mathbb{R}_{\geq 0} \mapsto f(d) \subseteq \mathbb{R}_{\geq 0}$). Linear timed sets form a restricted class of timed sets, which can be defined as $f\colon d \in \mathbb{R}_{\geq 0} \mapsto \bigcup_i (E_i + d) \cap [r_i; +\infty)$, with $E_i \subseteq \mathbb{R}$ and $r_i \in \mathbb{Q}_{\geq 0}$ for all i (see Fig. 3 on page 3 for an example of a linear timed set). Linear timed sets are well-suited to represent sets of clock valuations for one-clock timed automata, and compute their evolution along time, which is needed to update state estimations, as we illustrate on the following example.

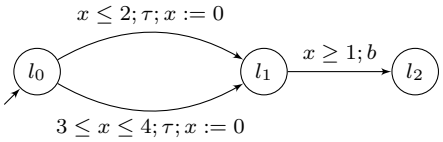

Fig. 1 A one-clock timed automaton where only the b-transition between l_1 and l_2 is observable.

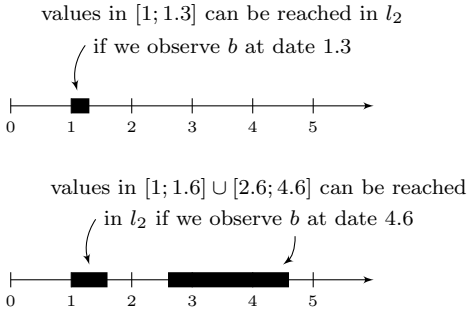

Fig. 2 Two sets (in location l_2) representing the reachable configurations in the automaton of Fig. 1 after observing transition b at dates 1.3 (left) and 4.6 (right).

Example 1 Consider the one-clock timed automaton of Fig. 1: in this automaton, the transition from l_1 to l_2 is observable (labelled with b), while the transitions from l_0 to l_1 are unobservable (labelled with the silent action τ).

Assume that this automaton starts from the initial configuration $(l_0, x = 0)$. As long as no b-action is observed, we have no way of knowing whether the automaton is in l_0 or l_1. Now, assume that a b-action takes place at time 1.3: then we know that one of the transitions from l_0 to l_1 has occurred; moreover, it cannot be the transition guarded with $3 \leq x \leq 4$, since only 1.3 time units have elapsed in total. One easily checks that, if we observe a b-transition at time 1.3, then the automaton is in state l_2 with $x \in [1; 1.3]$. Figure 2 (left) represents this set of valuations.

Similarly, if, starting from the initial configuration, we observe a b-transition at time 4.6, then both transitions from l_0 to l_1 may have taken place; in this situation, it can be checked that if the top transition has been taken, then the automaton can be in l_2 with $x \in [2.6; 4.6]$, while if the bottom transition has been taken, the automaton can be in l_2 with $x \in [1; 1.6]$. This set is represented on Fig. 2 (right).

For such an example, our algorithm would first compute the linear timed set

$$d \in \mathbb{R}_{\geq 0} \mapsto (([-4;-3] \cup [-2;0]) + d) \cap [0;+\infty),$$

corresponding to all clock valuations that can be obtained in l_1 after a delay d, as long as no transition has been observed. Since the b-transition can only take place if $x \geq 1$ (and does not reset clock x), the linear timed set reached when observing b is the intersection of this linear timed set with $[1;+\infty)$, namely:

$$d \in \mathbb{R}_{\geq 0} \mapsto (([-4;-3] \cup [-2;0]) + d) \cap [1;+\infty).$$

This linear timed set is represented at Fig. 3. It provides a way of representing, as a single object, all possible configurations that can be reached in l_2 right after observing transition b at time d, for any $d \in \mathbb{R}_{\geq 0}$. ◁

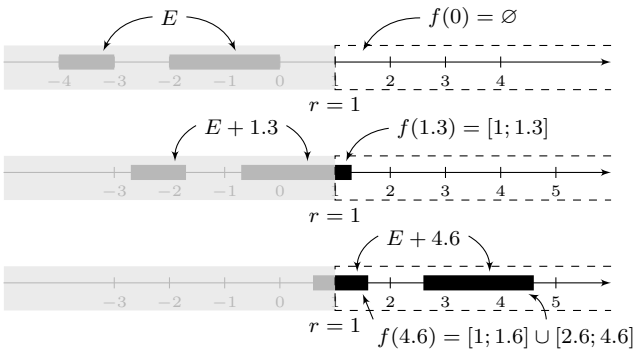

Fig. 3 The linear timed set $f\colon d \mapsto (([-4;-3] \cup [-2;0]) + d) \cap [1;+\infty)$ representing all reachable valuations that can be reached in l_2 when observing transition b at time d. The fact that $f(0) = \varnothing$ indicates that transition b cannot be taken at time 0.

In order to deal with all locations of a timed automaton, we use *markings*, which associate a set of valuations with each location of the automaton; similarly, *linear timed markings* associate a linear timed set with each location of the automaton: while sets and linear timed sets represent valuations, markings and linear timed markings represent sets of configurations.

Our algorithm consists in computing linear timed markings representing all configurations that can be reached from a given initial configuration after a given sequence of timed observations (alternations of observable transitions and delay transitions possibly including an arbitrary number of unobservable transitions). Given such a linear timed marking M, when an observation a is received at time t, we can easily compute the marking m containing all possible configurations in which the automaton can end up just after a a-transition is taken. From there, our algorithm computes the set of configurations that can be reached when

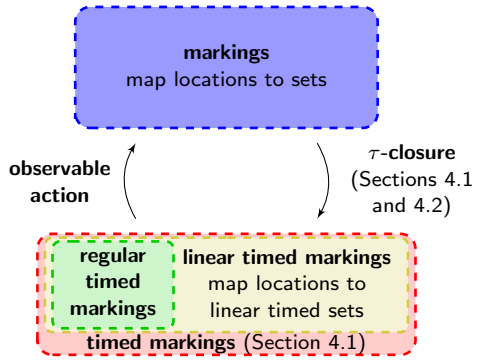

Fig. 4 Markings, linear timed markings, and related operations

time elapses (this is a linear timed marking), and the sets of configurations that can be reached by following any possible sequence of silent transitions. We prove that this can be effectively computed and finitely represented as a *regular* timed marking, which we call the τ-closure of m. Figure 4 is a graphical representation of those different concepts and operations.

Section 5 presents an implementation of our approach, which we compare to our implementation of Tripakis algorithm [27]. Thanks to our use of linear timed markings, computing the set of possible configurations after a delay transition can be performed very efficiently, contrary to the more brute-force approach of [27]. As our experiments show, our approach is more efficient for computing the effect of a delay transition by several orders of magnitude, while computing the effect of an observable transition is comparable.

Finally, in Section 6 we start extending our technique to timed automata with an arbitrary number of clocks: we prove in particular that linear timed markings are still sufficient to represent the configurations that can be reached after a given sequence of timed observations. But we have not been able to extend our notion of *regular* timed markings to a notion that could be used to properly represent and compute all reachable configurations in the n-clock setting; this is part of our future work.

Besides Sections 6 and 4.3, which are completely new, this version differs from the RV'18 version [11] by the inclusion of all the proofs, and the addition of a lot more explanations and intuitions.

Related works. State estimation is a classical challenge in timed automata. One of the main reason is that timed automata cannot in general be determinized [28, 16] and that silent transitions strictly increase their expressivity [5]. Thus a lot of methods usually relying on determinization (*e.g.* testing, runtime verification, diagnosis and to some extent model checking), or efficiently keeping track of the possible configurations [18], had to

come with other ways to estimate the current state of a system.

For example, Baier, Bertrand et al. have proposed a method to construct a deterministic timed tree from a timed automaton, with the limitation that the tree may be infinite [2].

Another approach is to impose some *restrictions* on the constructed model. Bouyer, Chevalier and D'Souza studied a restricted setting, only looking for diagnosers under the form of deterministic timed automata with limited resources [9]. Similarly, Krichen et al. have proposed to build a deterministic timed automaton from a non-deterministic one, by fixing a set of clocks and possibly making some approximations in the behaviour [23]. Stainer et al. later generalized and improved this approach using games [8]. Both works have then been used for off-line test generation from timed automata models [23, 7, 19].

One way to circumvent the problem is to use *online* algorithms and to update the set of possible states on the fly. Tripakis proposed the construction of a diagnoser as an online algorithm that keeps track of the possible states and zones the system can be in after each event (or after a sufficiently-long delay), which requires heavy online computation and is hardly usable in practice [27]. A similar method is used in UPPAAL TRON [21] and in IF TTG [23] to realize online testing of timed input output systems.

Finally, *automata over timed domains* [10], a larger determinizable class of models comprising timed automata has been created to offer a determinization procedure, but with a great expressivity - and hence complexity. This work is most tied with [10] from which automata over timed domains are borrowed, and [27] for the idea of computing sets of states after some finite execution -although we use a precomputation by opposition to Tripakis's fully online method.

2 Preliminaries

In this part (focusing on one-clock timed automata), we heavily use sets and intervals of reals with rational bounds, and especially unbounded ones. We first introduce these (and more generally notations for sets of reals with bounds in any subset of $\mathbb{R}$), and then the model of timed automata and related notions.

2.1 Sets and intervals of real

Let E and F be two subsets of $\mathbb{R}$, we define $E+F = \{e+f \mid e \in E,\ f \in F\}$ and $E-F = \{e-f \mid e \in E,\ f \in F\}$. For $d \in \mathbb{R}_{\geq 0}$, we write $E+d$ (resp. $E-d$) as a shorthand for $E+\{d\}$ (resp. $E-\{d\}$), used to shift E forward (resp. backward) by d.

For any subset $\mathbb{K}$ of $\mathbb{R}$, we write $\mathcal{I}_{\mathbb{K}}$ for the set of intervals of $\mathbb{R}$ with bounds in $\mathbb{K} \cup \{-\infty, +\infty\}$, and $\mathcal{I}_{\mathbb{K}_{\geq 0}}$ for the set of intervals with bounds in $\mathbb{K}_{\geq 0} \cup \{+\infty\}$.

For $r \in \mathbb{K}$, we define the following sets of $\mathcal{I}_{\mathbb{K}}$:

$$\uparrow\!\!\!\downarrow r = [r; +\infty) \qquad \uparrow r = (r; +\infty)$$
$$\downarrow r = (-\infty; r) \qquad \downarrow\!\!\!\uparrow r = (-\infty; r].$$

We write $\widehat{\mathbb{K}}_{\geq 0} = \{\uparrow\!\!\!\downarrow r, \uparrow r \mid r \in \mathbb{K}_{\geq 0}\}$ for the set of upward-closed intervals in $\mathcal{I}_{\mathbb{K}}$; in the sequel, elements of $\widehat{\mathbb{K}}_{\geq 0}$ are denoted with $\widehat{r}$. Similarly, $\underset{\sim}{\mathbb{K}}_{\geq 0} = \{\downarrow\!\!\!\uparrow r, \downarrow r \mid r \in \mathbb{K}_{\geq 0}\} \cup \{\mathbb{R}\}$, and we use notation $\underset{\sim}{r}$ for intervals in $\underset{\sim}{\mathbb{K}}_{\geq 0}$.

The following results are straightforward, and will be useful in the sequel:

Lemma 2 *Let $v \in \mathbb{K}_{\geq 0}$, $\widehat{r}$ and $\widehat{s}$ in $\widehat{\mathbb{K}}_{\geq 0}$, and $\underset{\sim}{t}$ in $\underset{\sim}{\mathbb{R}}_{\geq 0}$. Then*

- *if $v \notin \widehat{r}$, then $\widehat{r} \subseteq \uparrow v$;*
- *$\widehat{r} \cap \widehat{s} \in \{\widehat{r}, \widehat{s}\}$;*
- *if $\underset{\sim}{t}$ intersects both $\widehat{r}$ and $\widehat{s}$, then $\widehat{r} \cap \widehat{s} \cap \underset{\sim}{t} \neq \varnothing$.*

2.2 One-clock timed automata

Definition 3 Let Σ be a finite alphabet. A *one-clock timed automaton* over Σ is a tuple $\mathcal{A} = (L, \{l_0\}, \mathcal{C}, \Delta, \mathcal{F})$, where L is a finite set of locations, $l_0 \in L$ is the initial location, $\mathcal{C}$ is a set with a *unique* clock usually denoted x and $\Delta \subseteq L \times \mathcal{I}_{\mathbb{Q}_{\geq 0}} \times \Sigma \times 2^{\mathcal{C}} \times L$ is the set of transitions, $\mathcal{F} \subseteq L$ is a set of final states.

We usually call *valuation* a function $v\colon \mathcal{C} \to \mathbb{R}_{\geq 0}$ associating to the clock a non-negative value. In the sequel, we often identify valuations $v \in \mathbb{R}_{\geq 0}^{\mathcal{C}}$ and the real values $v(x) \in \mathbb{R}_{\geq 0}$ they return.

Let $\mathcal{A} = (L, \{l_0\}, \Delta, \mathcal{C}, \mathcal{F})$ be a one-clock timed automaton. A *configuration* of $\mathcal{A}$ is a pair $(l, v) \in L \times \mathbb{R}_{\geq 0}$. The semantics of $\mathcal{A}$ can be defined in terms of an infinite transition system whose states are the configurations of $\mathcal{A}$, in which there is a transition from (l, v) to (l', v') when

- $l' = l$ and $v' \geq v$: in that case, we write $(l, v) \xrightarrow{d} (l, v')$, with $d = v' - v$, for such *delay transitions* (notice that we have no invariants);
- there is a transition $e = (l, G, a, X, l')$ s.t. $v \in G$ and $v' = v$ if $X = \varnothing$, and $v' = 0$ otherwise (*i.e.* $X = \mathcal{C}$): for those *action transitions*, we write $(l, v) \to_e (l', v')$.

We write $\Delta_\varnothing$ for the set of *non-resetting* transitions, i.e., having $\varnothing$ as their fourth component, and $\Delta_\mathcal{C}$ for the complement set of *resetting* transitions.

For a transition $e = (l, G, a, X, l')$, the guard G is an interval, which can be written in a unique way as the intersection of an element $\widehat{e} \in \widehat{\mathbb{Q}}_{\geq 0}$ and an element $\underset{\sim}{e} \in \underset{\sim}{\mathbb{Q}}_{\geq 0}$. In the sequel, the guard of a transition e will often be written $\widehat{e} \cap \underset{\sim}{e}$. We write $\mathsf{src}(e) = l$ and $\mathsf{tgt}(e) = l'$, and $\mathsf{lab}(e) = a \in \Sigma$. We extend these definitions to sequences of transitions $w = (e_k)_{0 \leq k < n}$ as $\mathsf{src}(w) = \mathsf{src}(e_0)$, $\mathsf{tgt}(w) = \mathsf{tgt}(e_{n-1})$, and $\mathsf{lab}(w) = (\mathsf{lab}(e_k))_{0 \leq k < n}$.

We define runs of timed automata as finite sequences of transitions, with a strict alternation between delay transitions and action transitions; in the sequel, we write $(l, v) \xrightarrow{d}_w (l', v')$ when such a sequence exists following the sequence of action transitions $w \in \Delta^*$ with total duration d. Formally, let w be a sequence $(e_{2k+1})_{0 \leq 2k+1 < n}$ of transitions of Δ. For any non-negative real $d \in \mathbb{R}_{\geq 0}$, we write $(l, v) \xrightarrow{d}_w (l', v')$ if there exists a finite sequence of configurations $(l_k, v_k)_{0 \leq k \leq n} \in (L \times \mathbb{R}_{\geq 0})^{n+1}$ and a finite sequence of delays $(d_{2k})_{0 \leq 2k < n} \in \mathbb{R}_{\geq 0}^{\lfloor n+1/2 \rfloor}$ such that $\sum_{0 \leq 2k < n} d_{2k} = d$, and $(l_0, v_0) = (l, v)$ and $(l_n, v_n) = (l', v')$, and for all $0 \leq k' < n$, $(l_{k'}, v_{k'}) \xrightarrow{d_{k'}} (l_{k'+1}, v_{k'+1})$ if k' is even and $(l_{k'}, v_{k'}) \to_{e_{k'}} (l_{k'+1}, v_{k'+1})$ if k' is odd. We write $(l, v) \to (l', v')$ when $(l, v) \xrightarrow{d}_w (l', v')$ for some $w \in \Delta^*$ and some $d \in \mathbb{R}_{\geq 0}$.

For any $\lambda \in \Sigma^*$ and any $d \in \mathbb{R}_{\geq 0}$, we write $(l, v) \overset{d}{\rightsquigarrow}_\lambda (l', v')$ whenever there exists a sequence of transitions w such that $\lambda = \mathsf{lab}(w)$ and $(l, v) \xrightarrow{d}_w (l', v')$. Notice that $(l, v) \overset{d}{\rightsquigarrow}_\varepsilon (l', v')$ (where ε is the empty word) corresponds to delay transitions (hence it must be $l = l'$). The untimed language $\mathcal{L}(\mathcal{A})$ of $\mathcal{A}$ is the set of words $\lambda \in \Sigma^*$ such that $(l_0, 0) \overset{d}{\rightsquigarrow}_\lambda (l', v')$ for some $l' \in \mathcal{F}$ and $d \in \mathbb{R}_{\geq 0}$.

In the sequel, we heavily use *markings*, which map locations of $\mathcal{A}$ to sets of clock valuations, thereby representing sets of configurations of $\mathcal{A}$. In our context, the set of markings of a one-clock timed automaton $\mathcal{A} = (L, \{l_0\}, \mathcal{C}, \Delta, \mathcal{F})$ is the set $\mathbb{M} = \{m \mid m \colon L \to \mathcal{P}(\mathbb{R}_{\geq 0})\}$ [1]. A marking $m \in \mathbb{M}$ represents the set of configurations $\{(l, v) \mid v \in m(l)\}$.

For any $a \in \Sigma$, we define the function $\mathbf{O}_a \colon \mathbb{M} \to \mathbb{M}$ by letting, for any $m \in \mathbb{M}$ and any $l' \in L$,

$$\mathbf{O}_a(m) \colon l' \in L \mapsto \{v' \in \mathbb{R}_{\geq 0} \mid \exists l \in L.\ \exists v \in m(l).\ (l, v) \overset{0}{\rightsquigarrow}_a (l', v')\}.$$

[1] For any set U, we write $\mathcal{P}(U)$ for the set of subsets of U.

Then $\mathbf{O}_a(m)$ is the marking representing the set of configurations that can be reached after observing an a-transition from the configurations represented by m. Similarly, for any $d \in \mathbb{R}_{\geq 0}$, we let

$$\mathbf{O}_d(m) \colon l' \in L \mapsto \{v' \in \mathbb{R}_{\geq 0} \mid \exists l \in L.\ \exists v \in m(l).\ (l, v) \overset{d}{\rightsquigarrow}_\varepsilon (l', v')\},$$

representing the configurations reached from m after observing a delay of d time units. Notice that $\mathbf{O}_d$ simply shifts all valuations by d.

2.3 Timed automata with silent transitions

So far, we have considered that all transitions of timed automata can be observed. In that case, for any $d \in \mathbb{R}_{\geq 0}$, the operation $\mathbf{O}_d$ can easily be computed, since it amounts to adding d to each item of the marking (in other terms, for any marking m, any state $l \in L$, and any $v \in \mathbb{R}_{\geq 0}$, we have $v \in m(l)$ if, and only if, $v + d \in \mathbf{O}_d(m)(l)$).

We extend this approach to timed automata with unobservable actions: we assume that Σ contains a special *silent letter* τ, whose occurrences are not *observable*. This requires changing the definition of lab: for $w \in \Delta^*$ and $e \in \Delta$, we now let

$$\begin{aligned} &\mathsf{lab}(\varepsilon) = \varepsilon \\ &\mathsf{lab}(w \cdot e) = \mathsf{lab}(w) && \text{if } \mathsf{lab}(e) = \tau \\ &\mathsf{lab}(w \cdot e) = \mathsf{lab}(w) \cdot \mathsf{lab}(e) && \text{if } \mathsf{lab}(e) \neq \tau \end{aligned}$$

Notice that $\mathsf{lab}(w) \in (\Sigma \setminus \{\tau\})^*$. Notice also that time still is observable: we know exactly how much time elapses between observable actions, and how much time has elapsed since the last observation.

Now $(l, v) \overset{d}{\rightsquigarrow}_\varepsilon (l', v')$ indicates a sequence of zero or more silent transitions in d time units; in that case we may have $l' \neq l$. The function $\mathbf{O}_d$ cannot be computed anymore by just shifting valuations by d. In its raw form, the function $\mathbf{O}_d$ can be obtained by the computation of the set of reachable configurations in a delay d by following (arbitrarily long sequences of) silent transitions. This is analogous to the method proposed by Tripakis [27] for keeping track of the set of all possible configurations the automaton can be in after observation $w \in \Sigma^*$ and delay d. In [27], the set of possible configurations is updated each time a new action is observed (or after a time out, if no new observations occur); besides being very costly, this approach may increase the delay for detecting the occurrence of faulty actions.

Example 4 Consider again the one-clock timed automaton of Fig. 1, and the (initial) marking m_0 which maps l_0 to the single valuation v_0 such that $v_0(x) = 0$, and locations l_1 and l_2 to the empty set. This marking corresponds to a single valuation.

Assume that we observe transition b after 1.3 time units. Obviously, the automaton must have taken one of the transitions from l_1 to l_2, but since they are unobservable, we cannot know when this occurred. Actually, the bottom transition requires $3 \leq x \leq 4$, so it cannot have been used during the first 1.3 time units. In the end, it is not hard to check that

$$\begin{aligned}\mathbf{O}_{1.3}(m_0)\colon\ & l_0 \mapsto \{1.3\}\\ & l_1 \mapsto [0; 1.3]\\ & l_2 \mapsto \varnothing\end{aligned}$$

The b-transition can be taken from the configurations in l_1 where $x \geq 1$. This amounts to applying $\mathbf{O}_b$ to the marking above, which results in a marking $m_{1.3,b}$ mapping l_0 and l_1 to the empty set, and l_2 to $[1; 1.3]$.

Now, what if, instead, we observe b at time 4.6? Computing $\mathbf{O}_{4.6}(m_0)$ can be done as above, but now taking both transitions between l_0 and l_1 into account. This results in

$$\begin{aligned}\mathbf{O}_{4.6}(m_0)\colon\ & l_0 \mapsto \{4.6\}\\ & l_1 \mapsto [0.6; 1.6] \cup [2.6; 4.6]\\ & l_2 \mapsto \varnothing\end{aligned}$$

Then taking transition b is similar to the previous situation: it corresponds to applying $\mathbf{O}_b$, which results in a marking $m_{4.6,b}$ mapping l_0 and l_1 to the empty set, and l_2 to $[1; 1.6] \cup [2.6, 4.6]$. We recover the markings represented on Fig. 2. ◁

The two following sections will be devoted to the computation of $\mathbf{O}_d$ (and $\mathbf{O}_a$). First Section 3 introduces the notion of *linear timed sets*, which we use to represent sets of valuations that evolve over time, and its subclass of *regular timed sets* which allows a finite representation in the case of one-clock time automata. In Section 4 we lift these notions to sets of configurations and their evolution over time, by the notions of *linear timed markings* and *regular timed markings*. We define a τ-closure operator on linear timed markings, and, show that in the case of one-clock timed automata, it can be efficiently computed using regular timed markings, at the expense of some precomputations.

3 Regular timed sets

In this section, we define the notion of *linear timed set* to represent sets of clock valuations and their evolution over time. Most important to us is the subclass of *regular timed sets*, that uses regularity (as we will define just below) to ensure a finite representation, and will be key to efficiency in the computation of closure, as will be seen in the next section.

3.1 Regular unions of intervals

Before defining linear and regular timed sets, we introduce our notion of regularity for sets.

Definition 5 A *regular union of intervals* is a 4-tuple $R = (I, J, p, q)$ where I and J are finite unions of intervals in $\mathcal{I}_{\mathbb{Q}}$ (*e.g.* intervals of $\mathbb{R}$ with bounds in $\mathbb{Q} \cup \{-\infty, +\infty\}$), $p \in \mathbb{Q}_{>0}$ is the period, and $q \in \mathbb{N}$ is the offset. It is required that $J \subseteq (-p; 0]$ and $I \subseteq \mathbb{1}(-q \cdot p)$.

The regular union of intervals $R = (I, J, p, q)$ represents the subset of $\mathcal{I}_{\mathbb{Q}}$ $\mathcal{S}(R) = I \cup \bigcup_{k=q}^{+\infty} (J - k \cdot p)$ (where $J - k \cdot p$ is the interval obtained by shifting J by $-k \cdot p$).

Example 6 Fig. 5 shows an example of a regular union of intervals. There, the period is 1, the offset is 3, and the sets I and J are as displayed on the figure. ◁

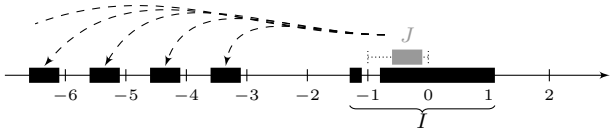

Fig. 5 Example of a regular union of intervals.

Regular unions of intervals enjoy the following properties:

Proposition 7 *Let R and R' be regular unions of intervals representing the sets of reals $\mathcal{S}(R)$ and $\mathcal{S}(R')$. Then one can define regular unions of intervals, denoted $\overline{R}$, $R \cup R'$ and $R + R'$ which represent respectively the sets $\overline{\mathcal{S}(R)}$, $\mathcal{S}(R) \cup \mathcal{S}(R')$, and $\mathcal{S}(R) + \mathcal{S}(R')$.*

Proof We now prove the result for $\overline{\mathcal{S}(R)}$: writing $\mathcal{S}(R) = I \cup \bigcup_{k=q}^{+\infty} J - k \cdot p$, thanks to the constraints imposed on I and J, we have

$$\overline{\mathcal{S}(R)} = (\mathbb{1}(-q \cdot p) \setminus I) \cup \bigcup_{k=q}^{+\infty} ((-p; 0] \setminus J) - k \cdot p.$$

For the union we write $R = (I, J, p, q)$ and $R' = (I', J', p', q')$, so that

$$\mathcal{S}(R) = I \cup \bigcup_{k=q}^{+\infty} J - k \cdot p \quad \mathcal{S}(R') = I' \cup \bigcup_{k=q'}^{+\infty} J' - k' \cdot p'.$$

We can write $p = \frac{a}{b}$ and $p' = \frac{a'}{b'}$, where a, b, a' and b' are positive integers. Let

$$N = \min\{n \in \mathbb{N} \mid n \cdot b \cdot a' \geq q \text{ and } n \cdot b' \cdot a \geq q'\}.$$

Then we can write

$$\mathcal{S}(R) = \left(I \cup \bigcup_{k=q}^{N \cdot b \cdot a' - 1} J - k \cdot p \right) \cup \bigcup_{k=N}^{+\infty} \left[\left(\bigcup_{r=0}^{b \cdot a' - 1} J - r \cdot p \right) - k \cdot a \cdot a' \right].$$

and

$$\mathcal{S}(R') = \left(I' \cup \bigcup_{k=q'}^{N \cdot b' \cdot a - 1} J' - k \cdot p' \right) \cup \bigcup_{k=N}^{+\infty} \left[\left(\bigcup_{r=0}^{b' \cdot a - 1} J' - r \cdot p' \right) - k \cdot a \cdot a' \right].$$

Since $J \subseteq (-p; 0]$, we have $\bigcup_{r=0}^{b \cdot a' - 1} J - r \cdot p \subseteq (-b \cdot a' \cdot p; 0] = (a \cdot a'; 0]$, and similarly for J'. Taking the union of the equalities above, we get an expression of $\mathcal{S}(R) \cup \mathcal{S}(R')$ under the form $I'' \cup \bigcup_{k=N}^{+\infty}(J'' - k \cdot (a \cdot a'))$, which proves that $\mathcal{S}(R) \cup \mathcal{S}(R')$ can be represented as a regular union of intervals.

The proof for $\mathcal{S}(R) + \mathcal{S}(R')$ follows similar ideas: as for $\mathcal{S}(R) \cup \mathcal{S}(R')$, we first rewrite $\mathcal{S}(R)$ and $\mathcal{S}(R')$ in such a way that they have the same period. Hence we assume w.l.o.g. $R = (I, J, p, q)$ and $R' = (I', J', p, q')$. Since I and J are finite unions of intervals, we can write

$$\mathcal{S}(R) = \bigcup_{k_i} I_{k_i} \cup \bigcup_{k_j} (\bigcup_{k=q}^{+\infty} J_{k_j} - kp)$$

where I_{k_i} and J_{k_j} are intervals. Similarly for $\mathcal{S}(R')$. This way, we can simply consider the following cases, and apply our previous result for union to prove that we end up with a regular union of intervals:

- $I_{k_i} + I'_{k_{i'}}$;
- $I_{k_i} + (\bigcup_{k=q'}^{+\infty} J'_{k_{j'}} - kp)$;
- $(\bigcup_{k=q}^{+\infty} J_{k_j} - kp) + (\bigcup_{k=q}^{+\infty} J'_{k_{j'}} - kp)$.

The first case is trivial. The second and third cases are easy to handle. □

Notice that a regular union of intervals is *finitely representable*.

3.2 Linear and regular timed sets

We introduce *linear timed sets* as a way to represent sets of clock valuations (and eventually markings), and how they evolve over time. Informally, a linear timed set represents a mapping from the non-negative reals to the set of subsets of $\mathbb{R}_{\geq 0}$. We begin with the definition of atomic timed sets, which form a special case.

Definition 8 An *atomic timed set* is a pair $T = (E; \widehat{r})$ where $E \subseteq \mathbb{R}$ and $\widehat{r} \in \widehat{\mathbb{Q}}_{\geq 0}$. With such a pair $T = (E; \widehat{r})$, we associate a mapping $f_T \colon \mathbb{R}_{\geq 0} \to 2^{\mathbb{R}_{\geq 0}}$ defined as $f_T(d) = (E + d) \cap \widehat{r}$. The set $f_T(d)$ represents the actual valuations after d time units. We call the second component $\widehat{r}$ a *filter*.

A *linear timed set* T is a countable set $\{T_k \mid k \in K\}$ (sometimes also denoted with $\bigsqcup_{k \in K} T_k$) of atomic timed sets. With such a linear timed set, we again associate a mapping $f_T \colon \mathbb{R}_{\geq 0} \to 2^{\mathbb{R}_{\geq 0}}$ defined as $f_T(d) = \bigcup_{k \in K} f_{T_k}(d)$. A linear timed set is *finite* when K is. We write $\mathcal{T}(\mathbb{R})$ for the set of linear timed sets of $\mathbb{R}$.

For a timed set T after a delay d, we will often call *actual* valuations the elements of $f_T(d) = (E + d) \cap \widehat{r}$ by contrast to *potential* valuations *i.e.* elements of $(E + d) \setminus \widehat{r}$ that are not valuations for this particular delay, but model valuations that will appear for greater delays.

Example 9 Fig. 3 displays an example of an atomic timed set $T = (E; \uparrow 1)$, with $E = [-4; -3] \cup [-2; 0]$. The picture displays the sets $f_T(0) = \varnothing$, $f_T(1.3) = [1; 1.3]$, and $f_T(4.6) = [1; 1.6] \cup [2.6; 4.6]$.

Notice that those sets correspond to the markings reached in the situations of Example 1 (see Fig. 2). Actually, T can be used to represent *all* clock valuations that may be reached in state l_2 in the automaton depicted on Fig. 1 (starting from the initial configuration $(l_0, 0)$), depending on the date at which b is observed. ◁

Given two linear timed sets T and T', we write $T \sqsubseteq T'$ whenever $f_T(d) \subseteq f_{T'}(d)$ for all $d \in \mathbb{R}_{\geq 0}$. This is a pre-order relation; it is not anti-symmetric as for instance $(\{1\}; \uparrow 0) \sqsubseteq (\{1\}; \uparrow 1)$ and $(\{1\}; \uparrow 1) \sqsubseteq (\{1\}; \uparrow 0)$. We write $T \equiv T'$ whenever $T \sqsubseteq T'$ and $T' \sqsubseteq T$.

Clearly, cycles in timed automata may generate linear timed sets where the set E is infinite. Think of a self-loop silently resetting the clock whenever it reaches 1: the resulting linear timed set would be $(-\mathbb{N}; \uparrow 0)$. We will prove that for keeping track of all the configurations of any one-clock timed automaton, it is always sufficient to use finite unions of atomic timed sets in which the first component has a simple shape, namely that of a regular union of intervals.

Definition 10 A *regular timed set* is a finite timed set $T = \{(E_k; \widehat{r}_k) \mid k \in K\}$ such that for all $k \in K$, the set $E_k \subseteq \mathbb{R}$ is the image by $\mathcal{S}$ of a regular union of intervals $R_k = (I_k, J_k, p_k, q_k)$.

This defines an adequate structure for representing and manipulating sets of configurations of one-clock timed automata and their evolution over time. In the sequel, we extend linear timed sets into *linear timed markings*, explain how to compute them, and show that *regular* timed markings are (necessary and) sufficient for representing all reachable configurations of partially-observable one-clock timed automata.

4 Closure under delay and silent transitions

In this section, we fix a one-clock timed automaton $\mathcal{A} = (L, \{l_0\}, \mathcal{C}, \Delta, \mathcal{F})$ over alphabet Σ, assumed to contain a silent letter τ, and aim at computing the functions $\mathbf{O}_a$ for any $a \in \Sigma$ and $\mathbf{O}_d$ for any $d \in \mathbb{R}_{\geq 0}$. Computing $\mathbf{O}_a(m)$ for $a \in \Sigma$ is not very involved: for a given location $l' \in L$, for each location $l \in L$ and each transition e labelled with a with source l and target l', it suffices to intersect $m(l)$ with the guard $\widehat{e} \cap \underline{e}$, and add the resulting interval (or the singleton $\{0\}$ if the intersection is non-empty and e is a resetting transition) to $\mathbf{O}_a(m)(l')$, *i.e.*:

$$\mathbf{O}_a(m)(l') = \Big(\bigcup_{(l,\widehat{e}\cap\underline{e},a,r,l') \in \Delta_\varnothing} m(l) \cap (\widehat{e} \cap \underline{e}) \Big) \cup \Big(\bigcup_{(l,\widehat{e}\cap\underline{e},a,r,l') \in \Delta_\mathcal{C}} \big(m(l) \cap (\widehat{e} \cap \underline{e})\big)_{[\mathcal{C}\leftarrow 0]} \Big).$$

From now on, we only focus on computing $\mathbf{O}_d$, for $d \in \mathbb{R}_{\geq 0}$. For this, it is sufficient to only consider silent transitions of $\mathcal{A}$: we let $U = U_\mathcal{C} \uplus U_\varnothing$ be the subset of Δ containing exactly the transitions labelled with τ, partitioned into those transitions that reset the clock (in $U_\mathcal{C}$), and those that do not (in $U_\varnothing$). We write $\mathcal{A}_\tau$ for the restriction of $\mathcal{A}$ to silent transitions, and only consider that automaton in the sequel. In the following, we first describe the effect of (silent) transitions on markings and use it to define the τ-closure in Part. 4.1 and then propose an operator θ to explicitly compute this closure in Part. 4.2, using linear timed sets. Finally we prove that this computation and its result can be finitely represented and effectively computed with regular timed sets in Part. 4.4. This process is summarized in Fig. 6.

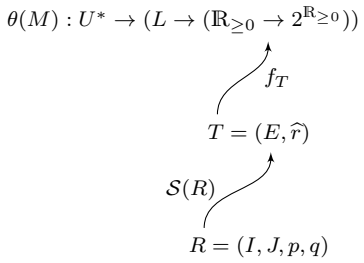

Fig. 6 Relation between the objects and their representations.

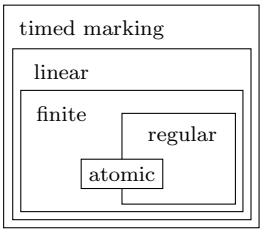

Fig. 7 The different classes of timed markings.

4.1 Linear timed markings and their τ-closure

We use markings to represent sets of configurations; in order to compute $\mathbf{O}_d$, we need to represent the evolution of these sets over time. For this, we introduce *timed markings*. A timed marking is a mapping $M \colon L \to (\mathbb{R}_{\geq 0} \to 2^{\mathbb{R}_{\geq 0}})$. For any delay $d \in \mathbb{R}$, we may (abusively) write $M(d)$ for the marking represented by M after delay d (so that for any $l \in L$ and any $d \in \mathbb{R}_{\geq 0}$, both notations $M(d)(l)$ and $M(l)(d)$ represent the same subset of valuations in $\mathbb{R}_{\geq 0}$).

For any $l \in L$ and any $d \in \mathbb{R}_{\geq 0}$, $M(l)(d)$ is intended to represent all clock valuations that can be obtained in l after a delay of d time units from the marking $M(0)$.

A special case of timed marking are those timed markings M that can be defined using linear timed sets, *i.e.*, for any l, $M(l)$ is a mapping f_T for some linear timed set T; timed markings of this kind will be called *linear timed markings* in the sequel. Atomic (resp. finite, regular) timed markings are linear timed markings whose values can be defined using atomic (resp. finite, regular) timed sets (we may omit to mention linearity in these cases to alleviate notations). The structure of these different classes of timed markings can be found in Fig. 7. As we prove below, regular timed markings are expressive enough to represent how markings evolve over time in one-clock timed automata.

Definition 11 We define the union of timed markings M_1 and M_2 as the timed marking such that $M_1 \cup M_2(l)(d) = M_1(l)(d) \cup M_2(l)(d)$. The intersection of two timed markings is defined similarly.

Two timed markings are said *equivalent* when for all locations l and delays d, $M_1(l)(d) = M_2(l)(d)$. We write $M_1 \equiv M_2$ when this is the case.

Proposition 12 *The union of two linear (resp. finite, regular) timed markings is a linear (resp. finite, regular) timed marking.*

Proof This can be seen directly by the stability of timed sets and regular unions of intervals by union, which comes respectively by definition and Prop. 7. □

As we prove below, regular timed markings are expressive enough to represent how markings evolve over time in one-clock timed automata.

We use (linear) timed markings to dispose of a representation in the form of *linear timed sets*, that can store both *actual* valuations, and *potential* valuations that do not exist yet but will exist after some time elapses, as displayed for example in Fig. 3. This representation has the advantage over markings to be time-independent as a single timed marking represents the set of valuations for all possible delays.

With any marking m, we associate a linear timed marking, which we write $\overrightarrow{m}$, defined as $\overrightarrow{m}(l)(d) = \{v + d \mid v \in m(l)\}$. This timed marking is linear since it can be defined *e.g.* as $\overrightarrow{m}(l) = f_{(m(l);\uparrow 0)}$. It can be used to represent all clock valuations that can be reached from marking m after any delay $d \in \mathbb{R}_{\geq 0}$, without taking any transition.

To go further, we define the effect of a sequence $w \in U^*$ of silent transitions on a marking m as the following timed marking m^w associating to each delay the marking of reachable configurations:

$$m^w \colon l' \mapsto d \mapsto \{v' \in \mathbb{R}_{\geq 0} \mid \exists l \in L. \\ \exists v \in m(l).\ (l, v) \xrightarrow{d}_w (l', v')\}.$$

The set $m^w(l')(d)$ corresponds to all configurations reachable in l' after reading w from a configuration in m with a delay of exactly d time units. By definition of the transition relation $\to_w$, for $m^w(d)$ to be non-empty, w must be a sequence of consecutive transitions. Notice that $m^\varepsilon = \overrightarrow{m}$ as one would expect [2] (and hence is linear). Going further, we can define the τ-closure of m as the timed marking m^τ such that $m^\tau(l)(d) = \bigcup_{w \in U^*} m^w(l)(d)$. By definition of $\mathbf{O}_d$, for any delay $d \in \mathbb{R}_{\geq 0}$, we have $\mathbf{O}_d(m) = m^\tau(d)$ for any marking m.

This formalism is easily extended to timed markings by taking for a timed marking M and a sequence of silent transitions $w \in U^*$:

$$M^w \colon l' \mapsto d \mapsto \{v' \in \mathbb{R}_{\geq 0} \mid \exists l \in L. \\ \exists d_0 \leq d.\ \exists v \in M(l)(d_0).\ (l, v) \xrightarrow{d-d_0}_w (l', v')\}$$

and defining $M^\tau(d)(l) = \bigcup_{w \in U^*} M^w(d)(l)$.

Remark 13 Notice that the definition of M^w differs from the one of m^w only by the addition of an initial delay d_0; this takes into account the fact that some potential configurations may become actual in M. By seeing $M(d)$ as a marking, we have

$$M^w \colon l' \mapsto d \mapsto \bigcup_{d_0 \leq d} M(d_0)^w(l')(d - d_0).$$

The following lemmas prove that this extension is self consistent and coherent with what was defined on markings. First, the effect of the empty sequence on a linear timed marking leaves it unchanged:

Lemma 14 *For any* linear *timed marking M, it holds $M^\varepsilon \equiv M$.*

Proof Since M is linear, for any l', $M(l')$ is a linear timed set, so that $M(l')(d) = \bigcup_{k \in \mathbb{N}} (E_k + d) \cap \widehat{r_k}$. On the other hand,

$$\begin{aligned} M^\varepsilon(l')(d) &= \{v' \in \mathbb{R}_{\geq 0} \mid \exists l \in L.\ \exists d_0 \leq d. \\ &\qquad \exists v \in M(l)(d_0).\ (l, v) \xrightarrow{d-d_0}_\varepsilon (l', v')\} \\ &= \{v' \in \mathbb{R}_{\geq 0} \mid \exists d_0 \leq d. \\ &\qquad \exists v \in M(l')(d_0).\ (l', v) \xrightarrow{d-d_0}_\varepsilon (l', v')\} \\ &= \{v' \in \mathbb{R}_{\geq 0} \mid \exists d_0 \leq d. \\ &\qquad \exists v \in M(l')(d_0).\ v' = v + (d - d_0)\} \\ &= \{v' \in \mathbb{R}_{\geq 0} \mid \exists d_0 \leq d.\ \exists k \in \mathbb{N}. \\ &\qquad \exists v \in (E_k + d_0) \cap \widehat{r_k}.\ v' = v + (d - d_0)\} \\ &= \{v' \in \mathbb{R}_{\geq 0} \mid \exists d_0 \leq d.\ \exists k \in \mathbb{N}. \\ &\qquad v' \in (E_k + d) \cap \widehat{r_k}\} \\ &= \bigcup_{k \in \mathbb{N}} (E_k + d) \cap \widehat{r_k}. \end{aligned}$$ □

Second, applying a sequence of silent transitions to a marking or to its corresponding linear timed marking yields the same result:

Lemma 15 *For any marking m and any sequence $w \in U^*$, it holds $(\overrightarrow{m})^w \equiv m^w$.*

Proof For any d and l', we have

$$\begin{aligned} (\overrightarrow{m})^w(d)(l') &= \{v' \in \mathbb{R}_{\geq 0} \mid \exists l \in L.\ \exists d_0 \leq d. \\ &\qquad \exists v'' \in \overrightarrow{m}(d_0)(l).(l, v'') \xrightarrow{d-d_0}_w (l', v')\} \\ &= \{v' \in \mathbb{R}_{\geq 0} \mid \exists l \in L.\ \exists d_0 \leq d. \\ &\qquad \exists v \in m(l).(l, v) \xrightarrow{d_0}_\varepsilon (l, v'') \xrightarrow{d-d_0}_w (l', v')\} \end{aligned}$$

[2] Remember that in m^ε, ε correspond to the empty sequence of transitions.

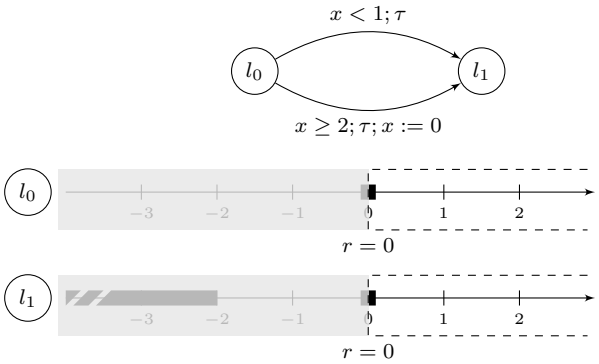

Fig. 8 A silent timed automaton and its reachable configurations.

On the other hand,

$$m^w(d)(l') = \{v' \in \mathbb{R}_{\geq 0} \mid \exists l \in L.\ \exists v \in m(l).\ (l, v) \xrightarrow{d}_w (l', v')\}.$$

It follows that any $v' \in (\overrightarrow{m})^w(d)(l')$ is in $m^w(d)(l')$, since $(l, v) \xrightarrow{d_0}_\varepsilon (l, v'') \xrightarrow{d-d_0}_w (l', v')$ implies $(l, v) \xrightarrow{d}_w (l', v')$. Conversely, any $v' \in m^w(d)(l')$ is in $(\overrightarrow{m})^w(d)(l')$, since if $(l, v) \xrightarrow{d}_w (l', v')$, then taking $d_0 = 0$, we have $(l, v) \xrightarrow{d_0}_\varepsilon (l, v) \xrightarrow{d-d_0}_w (l', v')$. □

This result can be generalized in the following way, linking even more the operations m^w (defined on markings) and M^w (defined on linear timed markings):

Corollary 16 *For any marking m and any two sequences of silent transitions w_1 and w_2 in U^*, it holds $(m^{w_1})^{w_2} \equiv m^{w_1 \cdot w_2}$.*

Proof For any m, w_1 and w_2:

$$\begin{aligned}(m^{w_1})^{w_2} &= ((\overrightarrow{m})^{w_1})^{w_2} && \text{(by lemma 15)}\\ &= (\overrightarrow{m})^{w_1 \cdot w_2} && \text{(by definition)}\\ &= m^{w_1 \cdot w_2} && \text{(by lemma 15.)}\end{aligned}$$

□

Finally, we formally define what we will consider as τ-closures in the sequel:

Definition 17 Let M be a timed marking. A timed marking N is a *τ-closure* of M if $N \equiv M^\tau$. The timed marking M is said *τ-closed* if it is a τ-closure of itself.

Our aim in this section is to compute (a finite representation of) a τ-closure of any given initial marking (defined using regular unions of intervals).

Example 18 Consider the (silent) timed automaton of Fig. 8. The initial configuration can be represented by the marking m defined as $m(l_0) = \{0\}$ and $m(l_1) = \varnothing$,

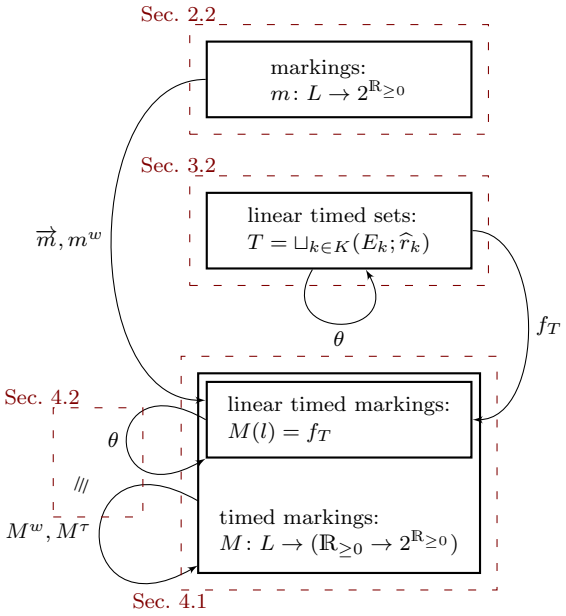

Fig. 9 Link between the different objects and plan of the paper.

corresponding to the single configuration $\{(l_0, x = 0)\}$. It gives rise to a timed marking $\overrightarrow{m}$ defined as $\overrightarrow{m}(l_0) = f_{(\{0\};\uparrow 0)}$ and $\overrightarrow{m}(l_1) = f_{(\varnothing;\uparrow 0)}$. Write M for this timed marking; M is not closed under silent-transitions, as for instance configuration $(l_1, 0)$ is reachable; however, this configuration cannot be reached after any delay: it is only reachable after delay 0, or after a delay larger than or equal to 2 time units. In the end, it can be checked that a *τ-closed* timed marking for this automaton is $M^\tau(l_0) = M(l_0)$, and $M^\tau(l_1) = ((-\infty; -2] \cup [0; 0]; \uparrow 0)$. ◁

4.2 Computing τ-closures

In this subsection, we show how to compute τ-closures. For this, we rely on linear timed set as a mean to represent $M(l)$ for the timed markings M and locations l encountered during the operation.

We will define an operator θ computing the effect of a silent transition on a linear timed set, and then extend it to sequences and languages of silent transitions on one hand and to linear timed markings on the other hand.

We will also prove that this operator corresponds to the semantic operations on linear timed markings M^w and M^τ used to compute the τ-closure.

The approach we have followed so far is summarized in Fig. 9, with links to the different sections.

Core to the definition of the θ operator is the *gauge*, that computes the effect of a resetting transition on a set of (potential) valuations.

Definition 19 Let E and G be two subsets of $\mathbb{R}$. We define their *gauge* as the set $E \ltimes G = (E - G) \cap \mathbb{R}_{\leq 0}$

This operation can be characterised in the following ways:

Proposition 20 *Let E and G be two subsets of $\mathbb{R}$. Then*

$$\begin{aligned} E \ltimes G &= \{-d \mid d \in \mathbb{R}_{\geq 0} \wedge (G - d) \cap E \neq \varnothing\} \\ &= \{-d \mid d \in \mathbb{R}_{\geq 0} \wedge (E + d) \cap G \neq \varnothing\} \end{aligned}$$

Proof We observe that

$$\begin{aligned} E \ltimes G &= \{e - g \mid e \in E,\ g \in G \text{ s.t. } e \leq g\} \\ &= \{-d \mid d \in \mathbb{R}_{\geq 0} \wedge \exists e \in E.\ \exists g \in G.\ d = g - e\} \\ &= \{-d \mid d \in \mathbb{R}_{\geq 0} \wedge \exists g \in G.\ g - d \in E\} \\ &= \{-d \mid d \in \mathbb{R}_{\geq 0} \wedge (G - d) \cap E \neq \varnothing\}. \end{aligned}$$

The other equality is proven similarly. □

These characterisations can be read as "$E \ltimes G$ is the inverse of the set of delays after which valuations in E satisfy the guard G". It corresponds to the intuition that after waiting such a delay, the transition can be taken and as it is a resetting one, the clock value becomes 0.

Example 21 Consider the simple transition in Fig. 10, with guard $G = [1, 2]$. When entering in l_0 with a set of potential configurations corresponding to $E = \{0\} \cup [-3, -2]$ displayed on the right side of the figure, the result of $E \ltimes G$ is the set of configurations $[-5, -3] \cup [-2, -1]$. To see this, imagine a right shift of E, and consider the delays d during which the two intervals composing E meet G, *i.e.* $[1, 2]$ and $[3, 5]$. Each such d is a date at which a configuration $(l_1, x = 0)$ can be spawned (because the transition resets the clock); therefore, $-d$ is a potential valuation in l_1. In the end, $(E \ltimes G; \uparrow 0) = ([-5, -3] \cup [-2, -1]; \uparrow 0)$ is the linear timed set of all potential valuations in l_1 in this situation. ◁

In the following, we use the gauge to define the operator θ on linear timed sets and extend it to *linear* timed markings, while proving that it corresponds to the semantic operations M^w and M^τ defined on general timed markings. The following simple statements will be useful in the sequel:

Proposition 22 – *If $E \leq G$ (that is, if for any $e \in E$ and any $g \in G$, it holds $e \leq g$; this is the case in particular if $E \subseteq \mathbb{R}_{\leq 0}$ and $G \subseteq \mathbb{R}_{\geq 0}$), then $E \ltimes G = E - G$;*

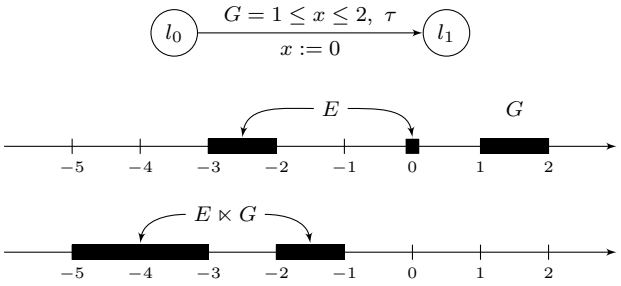

Fig. 10 Effect of the gauge operator.

– *if $E > G$ (*i.e.*, if for any $e \in E$ and any $g \in G$, it holds $e > g$), then $E \ltimes G = \varnothing$;*
– *If E and G are two intervals, then $E \ltimes G$ is an interval;*
– *If R is a regular union of intervals and G is an interval, then $\mathcal{S}(R) \ltimes G$ can be represented as a regular union of intervals noted $R \ltimes G$;*
– *If $G' \subseteq \mathbb{R}_{\geq 0}$, then $(E \ltimes G) \ltimes G' = (E \ltimes G) - G'$.*

Proof The first two claims are trivial from the definition of $E \ltimes G$. The third claim follows from the fact that $E - G$ is an interval if E and G are. The fourth claim follows from Prop. 7. Finally, the last claim is a consequence of the first one (because $E \ltimes G \subseteq \mathbb{R}_{\leq 0}$). □

We now define a mapping $\theta \colon \mathcal{T}(\mathbb{R}) \times U^* \to \mathcal{T}(\mathbb{R})$, intended to represent the linear timed set that is reached by performing sequences of silent transitions from some given linear timed set. We first consider atomic timed sets, and the application of a single silent transition.

Definition 23 For an atomic timed set $(E; \widehat{r})$ of $\mathcal{T}(\mathbb{R})$ and a transition $e = (l, \widehat{e} \cap \underline{e}, \tau, X, l')$ of U:

$$\theta((E; \widehat{r}), e) = \begin{cases} (\varnothing; \uparrow 0) & \text{if } \widehat{r} \cap \underline{e} = \varnothing \\ (E \cap \underline{e}; \widehat{r} \cap \widehat{e}) & \\ \qquad \text{if } \widehat{r} \cap \underline{e} \neq \varnothing \text{ and } X = \varnothing & \\ (E \ltimes (\widehat{r} \cap \widehat{e} \cap \underline{e}); \uparrow 0) & \\ \qquad \text{if } \widehat{r} \cap \underline{e} \neq \varnothing \text{ and } X = \mathcal{C} & \end{cases}$$

The intuition behind these three cases is as follows (see also Fig. 11 for a graphical illustration, and the proof of Lemma 26 for the formal arguments):

– if $\widehat{r} \cap \underline{e} = \varnothing$, then the transition cannot be taken: it means that the upper bound $\underline{e}$ of the guard is smaller than the smallest clock value in $\widehat{r}$ that can be reached;
– if $\widehat{r} \cap \underline{e} \neq \varnothing$, then there is some range of delays d such that $f_{(E;\widehat{r})}(d) \cap \widehat{e} \cap \underline{e}$ is not empty and the transition can be taken. Furthermore, the transition can be taken from any potential valuation in $E \cap \underline{e}$: for valuations in this set the transition will be possible after some delay, while valuations out of $\underline{e}$ will never go back in this set. Then after the transition:

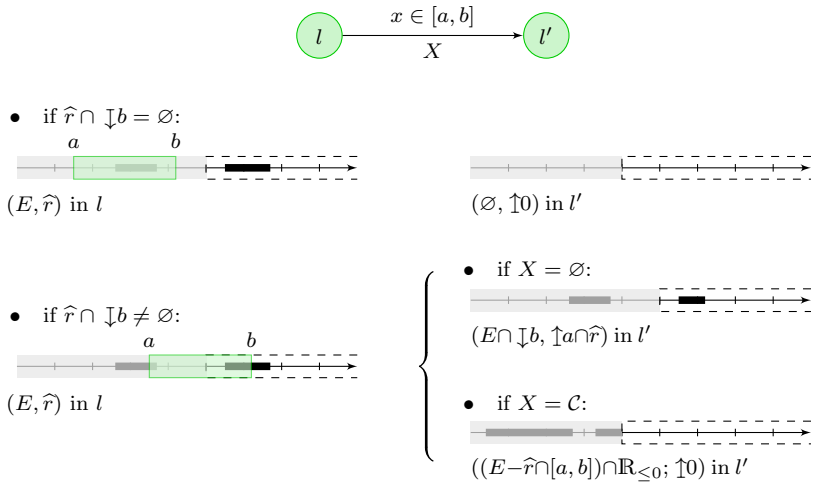

Fig. 11 Representation of the effect of silent transition $(l, [a,b], \tau, X, l')$ in three cases.

– if the transition does not reset the clock, then the value of the clock is not changed. Then $E \cap \underline{e}$ can represent the set of potential valuations. In order for a valuation to be actually reachable, it needs to be actual in $E; \widehat{r}$ (*i.e.* in $\widehat{r}$) and to have reached $\widehat{e} \cap \underline{e}$ (*i.e.* in $\widehat{e}$). Hence the timed set of reachable configurations can be represented by $(E \cap \underline{e}; \widehat{r} \cap \widehat{e})$;
– if the transition resets the clock, then the set of reachable values for the clock in l' after delay d_1 corresponds to the set of delays that can be spent in l' (after the clock reset), *i.e.*, the difference between d_1 and the delays d that can be spent in l before taking the transition. Those delays that can be spent in l before taking the transition can be seen to precisely correspond to $E \ltimes (\widehat{r} \cap \widehat{e} \cap \underline{e})$. Notice $\widehat{r} \cap \widehat{e} \cap \underline{e}$ corresponds to the set of actual valuations of $f_{(E;\widehat{r})}(d)$ satisfying the guard.

Example 24 Consider a linear timed set $T = (E = [-3,-2] \cup \{0\}, \uparrow n)$ with $n \in \mathbb{N}$ representing the configurations reachable in the state l_0 of the automaton in Fig. 10. If $n = 3$, then for all delays d, $f_T(d) \geq 3$ and it is clear that the transition cannot be taken. Hence, as expected, $\theta(T, l_0, \uparrow 1 \cap \downarrow 2, \tau, \{x\}, l_1) = \varnothing$ (this corresponds to the first case of the definition).

In contrast, if $n = 1$ then all configurations satisfying the guard can be reached after a fitting delay, and $\theta(T, l_0, \uparrow 1 \cap \downarrow 2, \tau, \{x\}, l_1) = (E \ltimes G, \uparrow 0)$ with $E \ltimes G = [-5,-3] \cup [-2,-1]$ as displayed on the figure. Notice that the new filter $\uparrow 0$ corresponds to the intuition: all positive clock values can be reached (after a certain delay), as the transition can be taken and it resets the clock. ◁

Definition 23 (cont.) We extend θ to sequences of transitions inductively as follows:

$$\theta((E;\widehat{r}), \varepsilon) = (E;\widehat{r})$$

and, for $w \in U^*$ and $e \in U$,

$$\theta((E;\widehat{r}), w{\cdot}e) = \begin{cases} \theta(\theta((E;\widehat{r}), w), e) & \text{if } \mathsf{tgt}(w) = \mathsf{src}(e) \\ (\varnothing; \uparrow 0) & \text{otherwise.} \end{cases}$$

Example 25 Consider a linear timed set $([-3,-1]; \uparrow 2)$, intended to represent the configurations that are reachable in a state l: then after 3 time units, only configuration $(l, 2)$ can be observed, and after 4.5 time units, the set of reachable configurations in l is $\{(l,x) \mid 2 \leq x \leq 3.5\}$.

Now, assume that there is a resetting transition from l to l', guarded with $x \in [1;4]$. The set of configurations reachable in l' (originating from the above linear timed set and transition) then is

$$\begin{aligned}([-3;-1] \ltimes \uparrow 2 \cap \uparrow 1 \cap \downarrow 4; \uparrow 0) &= ([-3;-1] \ltimes [2;4]; \uparrow 0) \\ &= ([-7;-3]; \uparrow 0).\end{aligned}$$

In particular, assuming we depart from linear timed set $([-3;-1]; \uparrow 2)$ in l, the transition can only take place between 3 and 7 time units. ◁

Definition 23 (cont.) We extend this definition to linear timed sets as follows:

$$\theta(\{T_k \mid k \in K\}, w) = \{\theta(T_k, w) \mid k \in K\}$$

The θ operator is now extended to all linear timed sets and sequences of transitions. We now prove that it indeed corresponds to the effect of taking transitions from a given timed set. For this we do not reason directly on the linear timed set but on their associated

functions f_T. The following lemma states that (informally) $\theta(T,w)$ corresponds to the effect of following w from T.

Lemma 26 *Let T be a linear timed set and $w \in U^*$. Then for any $d \in \mathbb{R}_{\geq 0}$ and any $v \in \mathbb{R}_{\geq 0}$,*

$$v \in f_{\theta(T,w)}(d) \;\Leftrightarrow\; \exists d_0 \in [0;d].\ \exists v' \in f_T(d_0).\ (\mathsf{src}(w), v') \xrightarrow{d-d_0}_w (\mathsf{tgt}(w), v).$$

Proof We carry the proof for the case where T is an atomic timed set. The extension to unions of atomic timed sets is straightforward.

The proof is in two parts: we begin with proving the result for $w = \varepsilon$, then for a single transition, and finally proceed by induction to prove the full result.

If $w = \varepsilon$, then $v \in f_\theta(T,w)(d)$ is equivalent to $v \in f_T(d)$, which entails the right-hand-side of the equivalence for $d_0 = d$. Conversely, if $v = v' + (d - d_0)$ for some $v' \in f_T(d_0)$, then, writing $T = (E; \widehat{r})$, we have $v' \in (E + d_0) \cap \widehat{r}$, so that $v \in (E+d) \cap \widehat{r}$, *i.e.* $v \in f_T(d)$.

Now, assume that w is a single transition $e = (l, \widehat{e} \cap \underline{e}, \tau, X, l')$, for which we assume (w.l.o.g.) that $\widehat{e} \cap \underline{e} \neq \varnothing$. In case T is empty (*i.e.* $E = \varnothing$), then also $\theta(T,e)$ is empty, and the result holds. We now assume that $T = (E; \widehat{r})$ is not empty (*i.e.* $E \neq \varnothing$), and consider three cases, corresponding to the three cases of the definition of $\theta(T,e)$:

- if $\widehat{r} \cap \underline{e} = \varnothing$, then $\theta(T,e) = (\varnothing; \uparrow 0)$. On the other hand, for any d_0 and any $v' \in f_T(d_0)$, it holds $v' \in \widehat{r}$, so that $v' \notin \underline{e}$, and the transition cannot be taken from that valuation (even after some delay). Hence both sides of the equivalence evaluate to false, and the equivalence holds.
- now assume that $\widehat{r} \cap \underline{e} \neq \varnothing$, and consider the case where e does not reset the clock. Write $\widehat{p} = \widehat{r} \cap \widehat{e}$, and pick $v \in f_{\theta(T,w)}(d) = f_{(E \cap \underline{e}; \widehat{p})}(d)$ (assuming one exists). Then $v \in (E + d) \cap (\underline{e} + d) \cap \widehat{p}$; this implies $\widehat{p} \cap \underline{e} \cap [v - d; v] \neq \varnothing$: indeed,
  - if $v \in \underline{e}$ or $v - d \in \widehat{p}$, then the result is trivial;
  - otherwise, $v \notin \underline{e}$ implies $\underline{e} \subseteq \downarrow v$, and $v - d \notin \widehat{p}$ implies $\widehat{p} \subseteq \uparrow v - d$, so that $\widehat{p} \cap \underline{e} \subseteq (v - d; v)$. Moreover, the fact that both $\widehat{r} \cap \underline{e}$ and $\widehat{e} \cap \underline{e}$ are non-empty implies that also $\widehat{p} \cap \underline{e} = \widehat{r} \cap \widehat{e} \cap \underline{e}$ is non-empty.

  Then for any v' in that set $\widehat{p} \cap \underline{e} \cap [v - d; v]$, letting $d_0 = v' - (v - d)$, we have $v' \in E + d_0$. In the end, $v' \in f_T(d_0)$, and $v' \in \widehat{e} \cap \underline{e}$, so that $(\mathsf{src}(e), v') \xrightarrow{d-d_0}_e (\mathsf{tgt}(e), v)$.

  Conversely, if $d_0 \in [0;d]$ and $v' \in f_T(d_0)$ exist such that $(\mathsf{src}(w), v') \xrightarrow{d-d_0}_w (\mathsf{tgt}(w), v)$, then $v' \in E + d_0 \cap \widehat{r}$, and for some $d_1 \leq d - d_0$, $v' + d_1 \in \widehat{e} \cap \underline{e}$. Then $v = v' + d - d_0$ since e does not reset the clock; also, since $v' \in E + d_0 \cap \widehat{r}$, we have $v \in (E + d) \cap \widehat{r}$; finally, from $v' + d_1 \in \widehat{e} \cap \underline{e}$, we get $v \in \widehat{e} + (d - d_0 - d_1) \subseteq \widehat{e}$ and $v \in \underline{e} + (d - d_0 - d1) \subseteq \underline{e} + d$. In the end, $v \in ((E + \underline{e}) + d) \cap (\widehat{r} \cap \widehat{e}) = f_{\theta(T,e)}(d)$.
- we finally consider the case where $\widehat{r} \cap \underline{e} \neq \varnothing$ and e resets the clock. In this case, $v \in f_{\theta(T,w)}(d)$ means that $v \in \uparrow 0$ and $v - d \in E \ltimes (\widehat{p} \cap \underline{e})$, which rewrites as $0 \leq v \leq d$ and $(E + d - v) \cap (\widehat{p} \cap \underline{e}) \neq \varnothing$ (by Prop. 22). Let $d_0 = d - v$. The property above entails that $0 \leq d_0 \leq d$, and that there exists some $v' \in (E + d_0) \cap (\widehat{p} \cap \underline{e})$, so that $0 \leq d_0 \leq d$, $v' \in f_T(d_0)$ and $(\mathsf{src}(e), v') \xrightarrow{d-d_0}_e (\mathsf{tgt}(e), v)$. Conversely, if those conditions hold, then for some $0 \leq d_1 \leq d - d_0$, we have $v' + d_1 \in \widehat{e} \cap \underline{e}$, and $v = d - (d_0 + d_1) \geq 0$ (remember that e resets the clock). Then $v' + d_1 \in E + (d_0 + d_1) \cap \widehat{r} \cap \widehat{e} \cap \underline{e}$, so that $-(d_0 + d_1) \in E \ltimes (\widehat{p} \cap \underline{e})$, and finally $v \in f_{\theta(T,w)(d)}$.

We now consider the case of $w \cdot e$, assuming that the result holds for $w \in U^+$. In case $\mathsf{tgt}(w) \neq \mathsf{src}(e)$, the result is trivial. Otherwise, first assume that $v' \in (\theta(T, w \cdot e))(d)$, and let $T' = \theta(T,w)$. Then $v' \in f_{\theta(T',e)}(d)$, thus there exist $0 \leq d_0 \leq d$ and $v \in f_{T'}(d_0)$ s.t. $(\mathsf{src}(e), v) \xrightarrow{d-d_0}_e (\mathsf{tgt}(e), v')$. Since $v \in f_{T'}(d_0)$, there must exist $0 \leq d_1 \leq d_0$ and $v'' \in f_T(d_1)$ such that $(\mathsf{src}(w), v'') \xrightarrow{d_0-d_1}_w (\mathsf{tgt}(w), v)$. We thus have found $0 \leq d_1 \leq d$ such that $(\mathsf{src}(w), v'') \xrightarrow{d-d_1}_{w \cdot e} (\mathsf{tgt}(e), v')$.

Conversely, if $(\mathsf{src}(w), v'') \xrightarrow{d-d_1}_{w \cdot e} (\mathsf{tgt}(e), v')$ for some $0 \leq d_1 \leq d$ and $v'' \in f_T(d_1)$, then we have $(\mathsf{src}(w), v'') \xrightarrow{d_0-d_1}_w (\mathsf{tgt}(w), v) \xrightarrow{d-d_0}_e (\mathsf{tgt}(e), v')$ for some $d_0 \in [d_1; d]$ and some v. In that case, we prove that $v \in f_{\theta(T,w)}(d_0)$: indeed, we have $d_1 \in [0; d_0]$, and $v'' \in f_T(d_1)$ such that $(\mathsf{src}(w), v'') \xrightarrow{d_0-d_1}_w (\mathsf{tgt}(w), v)$, which by induction hypothesis entails $v \in f_{\theta(T,w)}(d_0)$. Thus we have $d_0 \in [0; d]$ and $v \in f_{T'}(d_0)$, where $T' = \theta(T,w)$, such that $(\mathsf{tgt}(w), v) \xrightarrow{d-d_0}_e (\mathsf{tgt}(e), v')$, which means $v' \in f_{\theta(T',e)}(d)$, and concludes the proof. □

This results is the first link between θ and the semantics of timed automata, and the main technical result of the section, the following ones stemming from it. Thanks to this characterization of $\theta(T,w)$, we get:

Corollary 27 *For any sequence w of silent transitions, and any two equivalent linear timed sets T and T', the linear timed sets $\theta(T,w)$ and $\theta(T',w)$ are equivalent.*

This corollary ensures that we do not depend on the representation of the structure for our results since the mapping θ defined on linear timed sets preserves the equivalence of markings that they represent.

Finally, we extend θ to linear timed markings in the expected way:

Definition 23 (cont.) Given a linear timed marking M, and a sequence w of transitions, we let:

$$\theta(M,w)\colon \begin{array}{ll} l \mapsto f_{\theta(T_{M(\mathsf{src}(w))},w)} & \text{if } l = \mathsf{tgt}(w),\\ l \mapsto f_{(\varnothing;\uparrow 0)} & \text{otherwise.}\end{array}$$

with $T_{M(\mathsf{src}(w))}$ a timed set such that $f_T = M(\mathsf{src}(w))$.

Since $\theta(T_1 \sqcup T_2, w) = \theta(T_1, w) \sqcup \theta(T_2, w)$, we also have $\theta(M_1 \sqcup M_2, w) \equiv \theta(M_1, w) \sqcup \theta(M_2, w)$ when applied to linear timed markings. Then, we can extend the link between θ and the semantics to linear timed markings. This is the main result of the section, as it formally establishes the correspondence between the semantic computation of the effect of silent transitions and the operator θ that we build to compute it.

Theorem 28 *$\theta(M, w) \equiv M^w$ for all $w \in U^*$ and all linear timed marking M.*

Proof For $d \in \mathbb{R}_{\geq 0}$ and $l \in L$, we have

$$M^w(d)(l) = \{v \in \mathbb{R}_{\geq 0} \mid \exists l' \in L.\ \exists d_o \leq d.\ \exists v' \in M(d_0)(l').\ (l', v') \xrightarrow{d-d_0}_w (l, v)\}.$$

Hence clearly $M^w(l) \equiv f_{(\varnothing;\uparrow 0)}$ if $l \neq \mathsf{tgt}(w)$. When $l = \mathsf{tgt}(w)$, we have

$$M^w(d)(l) = \{v \in \mathbb{R}_{\geq 0} \mid \exists d_0 \leq d.\ \exists v' \in M(d_0)(\mathsf{src}(w)).\ (\mathsf{src}(w), v') \xrightarrow{d-d_0}_w (l, v)\}.$$

By Lemma 26, this corresponds to $\theta(M, w)(l)(d)$. □

Finally, we extend θ from sequences of transitions to languages.

Definition 29 For a timed marking M and a language $\mathcal{L} \subseteq U^*$, we let

$$\theta(M, \mathcal{L}) = \bigsqcup_{w \in \mathcal{L}} \theta(M, w)$$

and we note $\theta(M) = \theta(M, U^*)$.

From this definition and the previous statements, we immediately get:

Corollary 30 *For any linear timed marking M, it holds $\theta(M) \equiv M^\tau$.*

It follows that the τ-closure of any marking (in particular the initial marking) can be represented as a linear timed marking. However, this linear timed marking is currently defined as an infinite union over all sequences of consecutive silent transitions. We make the computation more effective (and representable) in the Section 4.4.

4.3 Necessity of regular timed sets

The next section will prove that regular timed sets are enough to express (and effectively compute) the set of configurations reached by a one-clock timed automaton. To complete this result, we prove that any regular timed marking can appear in the τ-closure of a one-clock timed automaton.

Proposition 31 *Given a regular timed set T, there exists a one-clock timed automaton $\mathcal{A}$ such that T represents the τ-closure of $\mathcal{A}$.*

Proof The proof is first made for an atomic regular timed set $T = (E; \widehat{r})$ defined using a regular union of intervals $R = (I, J, p, q)$. The generalization to finite unions of atomic regular timed sets is then proposed at the end. We prove this result by constructing a timed automaton. For this, we first explain how to encode intervals with bounds in $\mathbb{Q}_{\leq 0} \cup \{-\infty\}$, regular repetitions, finite unions, and how to add a filter $\widehat{r}$. Then we construct an automaton for T.

- Consider an interval I with bounds in $\mathbb{Q}_{\leq 0} \cup \{-\infty\}$. In the automaton represented in Fig. 12, we start from the initial (atomic regular) timed marking M_{init} associating $f_{(\{0\};\uparrow 0)}$ with the initial location l_0 and $f_{(\varnothing;\uparrow 0)}$ with any other location. The τ-closure $M^\tau_{init} = \theta(M_{init})$ of M_{init} associates $f_{(\{0\};\uparrow 0)}$ with l_0, and $f_{(I;\uparrow 0)}$ with l_1. Indeed, by definition of θ, the closure is computed as: $(\{0\} \ltimes -I \cap \uparrow 0; \uparrow 0) = (\{0\} \ltimes -I; \uparrow 0) = (I; \uparrow 0)$.

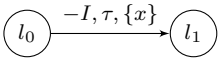

Fig. 12 An automaton whose closure in l_1 is $(I, \uparrow 0)$.

- Consider a rational $p \in \mathbb{Q}_{\leq 0}$, an integer q, an interval $J \in (-p; 0]$ with rational bounds, and the automaton depicted in Fig. 13. As argued in the previous case, starting from M_{init}, the effect of the transition $l_0 \to l_1$ through θ is represented by $(J - p \cdot q; \uparrow 0)$. Then it is easy to see that the τ-closure is represented by $(\bigcup_{k=q}^{\infty} J - k \cdot p; \uparrow 0)$ in l_1. Indeed, the effect of one application of the self-loop has the form $\cdot \ltimes \{p\}$, effectively repeating the pattern shifted by $-p$. The fixpoint then is $(\bigcup_{k=q}^{\infty} J - k \cdot p; \uparrow 0)$ in l_1.
- The general method to combine a finite number of atomic timed sets representing different behaviours ending in the same configuration is to regroup them in a finite timed set (*i.e.*, a finite collection of atomic timed sets). When all the atomic timed sets share the same constraint $\widehat{r}$, an equivalent method is to

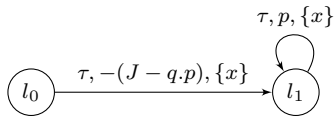

Fig. 13 An automaton whose closure in l_1 is $(\bigcup_{k=q}^{\infty} J - k.p; \uparrow 0)$.

take the explicit union of their first part. As argued in Prop. 7, when the atomic timed sets are regular, this yields an atomic regular timed set. A simple example of such unions is shown in Fig. 14.

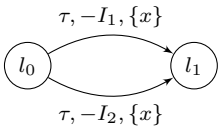

Fig. 14 An automaton having $(I_1 \cup I_2; \uparrow 0)$ as closure in l_1.

– Adding a stricter filter $\widehat{r}$ to a linear timed set through a transition is straightforward: it is the exact effect of a non-resetting transition of guard $\widehat{r}$.

Using these components, we can build an automaton using $T = (\mathcal{S}(R); \widehat{r})$ to encode the τ-closure of the initial timed marking. A possible example is depicted in Fig. 15 for $R = (\bigcup_{k=1}^{n_I} I_k, \bigcup_{k=1}^{n_J} J_k, p, q)$, where the dotted line is meant to represent the n_I and n_J edges. In this automaton, the closure can be represented by $(\mathcal{S}(R); \uparrow 0)$ in l_R and T is l_f.

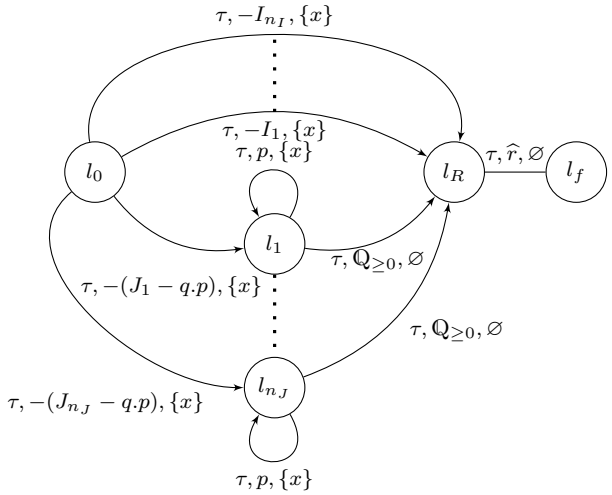

Fig. 15 An automaton using T to encode the closure in l_f.

It is clear to see that for a regular timed set $T = \{(E_k; \widehat{r}_k) \mid k \in K\}$, as K is by definition finite, building all components necessary to accept the $E_k; \widehat{r}_k$ with the same initial and final location yields a finite timed automaton with 1 clock requiring T to represent its τ-closure. □

Remark 32 Notice that we rely on the intervals defining a regular union of intervals having bounds in $\mathbb{Q}_{\leq 0}$, in order to define automaton guards with their additive inverse.

4.4 Finite representation of the closure

In Section 4.2, we demonstrated how the τ-closure of any linear timed marking (and hence of any marking) can be computed as a linear timed marking using the θ operator. To complete this result, we prove in this subsection that when starting from regular timed markings, the τ-closure can be computed with regular timed markings. This presents two main interests:

– First, it corresponds to the semantics of timed automata, as the initial timed marking is clearly regular and we have proved that any regular timed set can appear in a τ-closure (Prop. 31),
– second, linear timed markings rely on linear timed sets, which in general use infinite unions of sets of $\mathbb{R}$. In order to get an effective algorithm, we need a finite representation of sets, which regular unions of intervals provide.

The following theorem is the general result we prove in the following discussion. We call *language of transitions* of a timed automaton $\mathcal{A}$, noted $\mathcal{L}_{\Delta}(\mathcal{A})$, the language obtained by reading the name of the transitions instead of the labels.

Theorem 33 *Consider a timed automaton $\mathcal{A}$ and a regular language $\mathcal{L} \subseteq \mathcal{L}_{\Delta}(\mathcal{A})$ included in its language of transitions. Given a regular timed marking M, the timed marking $\theta(M, \mathcal{L})$ is regular.*

Thanks to this theorem and the fact that the timed marking corresponding to the initial valuation is regular, we can deduce the following facts:

Corollary 34 *Given a timed automaton $\mathcal{A}$ with τ transitions and its silent fragment A_τ:*

– *the initial (timed) marking of $\mathcal{A}_\tau$ is regular and*
– *the τ-closure of any regular (timed) marking is regular.*

This result entails that regular timed markings are indeed enough to represent and compute the τ-closures.

The remaining of this subsection is a proof of the previous Theorem 33. We first prove in the Paragraph 4.4.1 that the linear timed marking $\theta(M, \mathcal{L})$ is a *finite* timed marking, by discussing the set of possible filters. Then we prove that it is a regular timed marking in Paragraph 4.4.2.

4.4.1 Finiteness

Finiteness is achieved by separating the initial timed marking per location, and then proving that there is only a finite amount of possible filters.

Let M be a regular timed marking: then M can be written as the finite union of atomic regular timed markings $M_{(l,E,\widehat{r})}$, defined as $M_{(l,E,\widehat{r})}(l) = f_{(E;\widehat{r})}$ and $M_{(l,E,\widehat{r})}(l') = f_{(\varnothing;\uparrow 0)}$ for all $l' \neq l$. Thus $\theta(M,\mathcal{L})$ is the function associating to an input the union of the corresponding outputs of $\theta(M_{(l,E,\widehat{r})},\mathcal{L})$ and it suffices to compute the result for atomic regular timed markings $M_{(l,E,\widehat{r})}$.

We write $\theta_1((E;\widehat{r}),w) \subseteq \mathbb{R}$ and $\theta_2((E;\widehat{r}),w) \in \widehat{\mathbb{R}}_{\geq 0}$ for the first and second components of $\theta((E;\widehat{r}),w)$. Notice that $\theta_2((E;\widehat{r}),w)$ does not depend on E (so that we may denote it with $\theta_2(\widehat{r},w)$ in the sequel). In particular,

- $\theta_2((E;\widehat{r}),w) = \uparrow 0$ if $w \in U^* \times U_{\mathcal{C}}$ is a sequence of consecutive transitions ending with a resetting transition;
- $\theta_2((E;\widehat{r}),w) = \widehat{r} \cap \bigcap_{k<n} \widehat{e}_k$ if $w = e_1 \dots e_n \in {U_{\varnothing}}^*$ is a sequence of consecutive non-resetting transitions.

Letting $\mathbb{J}_{\widehat{r}} = \{\uparrow 0, \widehat{r}\} \cup \{\widehat{e} \mid e \in U_{\varnothing}\}$, it follows that $\theta_2((E;\widehat{r}),w) \in \mathbb{J}_{\widehat{r}}$ for any $(E;\widehat{r})$ and any w (because by Lemma 2, $\widehat{r} \cap \widehat{e}$ is either $\widehat{r}$ or $\widehat{e}$, for any $\widehat{r}$ and $\widehat{e}$ in $\widehat{\mathbb{R}}_{\geq 0}$). Hence $\theta(M_{(l,E,\widehat{r})})$, and thus also $\theta(M,\mathcal{L})$, can be written as a finite union of atomic timed markings as $\mathbb{J}_{\widehat{r}}$ is a finite set. By definition, $\theta(M,\mathcal{L})$ is thus a finite timed marking.

4.4.2 Regularity

In the following, we reason on each $M_{(l,E,\widehat{r})}$ separately. To prove regularity, we first introduce some more notations and state Lemma 35, that allows to rewrite θ using the properties of the gauge. Then we separate $\theta(M_{(l,E,\widehat{r})},\mathcal{L})$ as we did for M by separating first the ending locations and then the ending filters. For this we separate $\mathcal{L} = \bigcup_{l,l' \in L,\ \widehat{r},\widehat{r}' \in \mathbb{J}_{\widehat{r}}} [\mathcal{L}(l,l')]_{\widehat{r}}^{\widehat{r}'}$. The proof that $\theta(M_{(l,E,\widehat{r})}, [\mathcal{L}(l,l')]_{\widehat{r}}^{\widehat{r}'})$ is indeed regular is encapsulated in Lemma 36.

First, some more formalism:

- we let $\widehat{G_{id}} = \{\widehat{e} \mid e \in U_{\varnothing}\}$ and $\underline{G_{id}} = \{\underline{e} \mid e \in U_{\varnothing}\}$. We thus have $\mathbb{J}_{\widehat{r}} = \{\uparrow 0, \widehat{r}\} \cup \widehat{G_{id}}$;
- for $\widehat{r} \in \widehat{\mathbb{R}}_{\geq 0}$ and $e \in U$, we write $\Phi(\widehat{r},e)$ for the interval $\widehat{r} \cap \widehat{e} \cap \underline{e}$;
- we define a mapping $\mathcal{J}_{\widehat{r}} \colon U^* \to \mathbb{N}^{\mathbb{J}_{\widehat{r}} \times U_{\mathcal{C}}}$ that counts the number of occurrences of certain timing constraints at resetting transitions along a path: precisely, it is defined inductively as follows (where $\uplus$ represents addition of an element to a multiset):

$$\begin{aligned}
\mathcal{J}_{\widehat{r}}(\varepsilon) &= \{0\}^{\mathbb{J}_{\widehat{r}} \times U_{\mathcal{C}}} \\
\mathcal{J}_{\widehat{r}}(w \cdot e) &= \mathcal{J}_{\widehat{r}}(w) \uplus \{(\theta_2(\widehat{r},w),e)\} && \text{if } e \in U_{\mathcal{C}} \\
\mathcal{J}_{\widehat{r}}(w \cdot e) &= \mathcal{J}_{\widehat{r}}(w) && \text{if } e \in U_{\varnothing}.
\end{aligned}$$

The idea behind $\mathcal{J}_{\widehat{r}}$ is the following: consider a timed set $(E;\widehat{r})$, and a resetting silent transition guarded with $x \in \underline{e} \cap \widehat{e}$. Resetting clock x when it is in $\widehat{r} \cap \underline{e} \cap \widehat{e}$ amounts to substracting to x some value in that interval. The function $\mathcal{J}_{\widehat{r}}$ precisely counts the number of occurrences of each of those intervals along a sequence w of silent transitions. Since there is only one clock, the order of the transitions is not important. Lemma 35 formalizes this intuition:

Lemma 35 *Let $(E;\widehat{r})$ be an atomic timed set with $E \subseteq \mathbb{R}_{\leq 0}$, and $w \in U^*$. Then either $\theta_1((E;\widehat{r}),w) = \varnothing$, or*

$$\begin{aligned}
\theta_1((E;\widehat{r}),w) &= E - \sum_{J=(\widehat{g},e) \in \mathbb{J}_{\widehat{r}} \times U_{\mathcal{C}}} \mathcal{J}_{\widehat{r}}(w)(J) \times \Phi(\widehat{g},e) \\
&= \Big\{ x - \sum_{J \in \mathbb{J}_{\widehat{r}} \times U_{\mathcal{C}}} \mathcal{J}_{\widehat{r}}(w)(J) \cdot y_J \;\Big|\; x \in E \text{ and } \\
&\qquad y_J \in \Phi(J) \text{ for all } J \in \mathbb{J}_{\widehat{r}} \times U_{\mathcal{C}} \Big\}.
\end{aligned}$$

Proof The result is straightforward for $w = \epsilon$. Now, assume the result holds for some $w \in U^*$, and consider $w' = w \cdot e$.

Assuming that $\theta_1((E;\widehat{r}),w) \neq \varnothing$, we first consider the case where $e \in U_{\varnothing}$: then

$$\theta_1((E;\widehat{r}),w \cdot e) = \theta_1((E;\widehat{r}),w) \cap \underline{e}$$

(by Definition 23 of θ). Since $\theta_1((E;\widehat{r}),w) \subseteq \mathbb{R}_{\leq 0} \subseteq \underline{e}$, we have $\theta_1((E;\widehat{r}),w \cdot e) = \theta_1((E;\widehat{r}),w)$. Since $\mathcal{J}_{\widehat{r}}(w \cdot e) = \mathcal{J}_{\widehat{r}}(w)$ when $e \in U_{\varnothing}$, our result follows.

Now assume $e \in U_{\mathcal{C}}$: we have

$$\theta_1((E;\widehat{r}),w \cdot e) = \theta_1((E;\widehat{r}),w) \ltimes (\theta_2(\widehat{r},w) \cap \widehat{e} \cap \underline{e}).$$

Since $\theta_1((E;\widehat{r}),w) \subseteq \mathbb{R}_{\leq 0}$ and $\theta_2(\widehat{r},w) \subseteq \mathbb{R}_{\geq 0}$, Prop. 22 entails

$$\theta_1((E;\widehat{r}),w \cdot e) = \theta_1((E;\widehat{r}),w) - (\theta_2(\widehat{r},w) \cap \widehat{e} \cap \underline{e}).$$

Then, since $\Phi(\theta_2(\widehat{r},w),e) = \theta_2(\widehat{r},w) \cap \widehat{e} \cap \underline{e}$, substracting $\theta_2(\widehat{r},w) \cap \widehat{e} \cap \underline{e}$ precisely corresponds to the effect of adding $\{(\theta_2(\widehat{r},w),e)\}$ to the multiset $\mathcal{J}_{\widehat{r}}(w)$. □

Let $M_{(l,E,\widehat{r})}$ be an atomic regular timed marking. As regular timed markings are defined using linear timed sets, we extend the notations θ_1 and θ_2 to regular timed markings as the functions returning resp. mappings from locations to sets (that we want to define using regular unions of intervals), and elements of $\widehat{\mathbb{R}}_{\geq 0}$, such that for any regular timed marking M, we have $\theta(M,w) \colon l \mapsto$

$f_{(\theta_1(M,w)(l);\theta_2(M,w)(l))}$. For any state l' of $\mathcal{A}_\tau$, we let $\mathcal{L}(l,l')$ be the set of all sequences of consecutive transitions from l to l' in $\mathcal{A}_\tau$. Then

$$(M_{(l,E,\widehat{r})})^\tau = \bigsqcup_{l' \in L} \theta(M_{(l,E,\widehat{r})}, \mathcal{L}(l,l')).$$

Hence it suffices to prove that $\theta(M_{(l,E,\widehat{r})}, \mathcal{L}(l,l'))$ is regular.

For any set $\mathcal{L}$ of sequences of consecutive transitions, and for any $\widehat{r}$ and $\widehat{r}'$ in $\widehat{\mathbb{R}}_{\geq 0}$, we let $\mathcal{L}^{\widehat{r}'}_{\widehat{r}} = \{w \in \mathcal{L} \mid \widehat{r}' = \theta_2(\widehat{r}, w)\}$. One easily observes that for any $\widehat{r} \in \widehat{\mathbb{R}}_{\geq 0}$ and any $\mathcal{L}$, it holds $\mathcal{L} = \bigcup_{\widehat{r}' \in \mathbb{J}_{\widehat{r}}} \mathcal{L}^{\widehat{r}'}_{\widehat{r}}$, so that $\theta(M_{(l,E,\widehat{r})}, \mathcal{L}(l,l'))$ can be written as

$$\bigsqcup_{\widehat{r}' \in \mathbb{J}_{\widehat{r}}} l'' \mapsto f_{\left(\theta_1(M_{(l,E,\widehat{r})}, [\mathcal{L}(l,l')]^{\widehat{r}'}_{\widehat{r}})(l'');\widehat{r}'\right)}.$$

We can thus focus on $\theta_1(M_{(l,E,\widehat{r})}, [\mathcal{L}(l,l')]^{\widehat{r}'}_{\widehat{r}})$. The following key lemma entails that this set is a regular union of intervals:

Lemma 36 *Let $M_{(l,E,\widehat{r})}$ be an atomic regular timed marking. Let $\widehat{r}$ and $\widehat{r}'$ be two elements of $\widehat{\mathbb{R}}_{\geq 0}$, and $\mathcal{L} \subseteq \mathcal{L}(\mathcal{A}_\tau)$ be a* regular *language. Then $\theta_1(M_{(l,E,\widehat{r})}, \mathcal{L}^{\widehat{r}'}_{\widehat{r}})$ can be defined using a regular union of intervals.*

Proof The proof of this result is in two parts: we first express $\mathcal{L}^{\widehat{r}'}_{\widehat{r}}$ as the language of a finite automaton, and then—by a tedious proof—express $\theta_1(M_{(l,E,\widehat{r})}, \mathcal{L}^{\widehat{r}'}_{\widehat{r}})$ using regular unions of intervals.

Finite automaton for $\mathcal{L}^{\widehat{r}'}_{\widehat{r}}$. We assume that $\mathcal{L}$ is accepted by some automaton $\mathcal{B} = (L, \{l_0\}, \mathcal{C}, U, \mathcal{F})$ (in particular, $\mathcal{L}(l,l')$ would be obtained from $\mathcal{A}_\tau$ by imposing an initial state l and a single accepting state l'). In order to derive a finite automaton accepting $\mathcal{L}^{\widehat{r}'}_{\widehat{r}}$, we have to keep track of the value of θ_2 along runs. For this, we take the product of $\mathcal{B}$ with $\mathbb{J}_{\widehat{r}}$: we define the automaton $\mathcal{B}^{\widehat{r}'}_{\widehat{r}} = (L \times \mathbb{J}_{\widehat{r}}, \{(l_0, \widehat{r})\}, \mathcal{C}, U', \mathcal{F} \times \{\widehat{r}'\})$ over the extended alphabet $\widehat{\mathbb{R}}_{\geq 0} \times U \times \widehat{\mathbb{R}}_{\geq 0}$ (this will be useful for technical reasons), where

$$\begin{aligned} U' = \{&([l, \widehat{g}], \widehat{e} \cap \underbrace{e}, (\widehat{g}, e, \uparrow 0), \varnothing, [l', \uparrow 0]) \mid \\ &\qquad e = (l, \widehat{e} \cap \underbrace{e}, \varnothing, \tau, l') \in U \text{ and } \widehat{g} \cap \underbrace{e} \neq \varnothing\} \cup \\ &\{([l, \widehat{g}], \widehat{e} \cap \underbrace{e}, (\widehat{g}, e, \widehat{g}'), \mathcal{C}, [l', \widehat{g}']) \mid \\ &\qquad e = (l, \widehat{e} \cap \underbrace{e}, \mathcal{C}, \tau, l') \in U \text{ and } \widehat{g} \cap \underbrace{e} \neq \varnothing \\ &\qquad\qquad \text{and } \widehat{g}' = \widehat{g} \cap \widehat{e}\}. \end{aligned}$$

This automaton accepts words in $(\widehat{\mathbb{R}}_{\geq 0} \times U \times \widehat{\mathbb{R}}_{\geq 0})^*$. For any $\widehat{r} \in \widehat{\mathbb{R}}_{\geq 0}$, we define $\kappa_{\widehat{r}} \colon U \to (\widehat{\mathbb{R}}_{\geq 0} \times U \times \widehat{\mathbb{R}}_{\geq 0})$ as

$$\kappa_{\widehat{r}}(e) = \begin{cases} (\widehat{r}, e, \widehat{r} \cap \widehat{e}) & \text{if } e \in U_\varnothing \\ (\widehat{r}, e, \uparrow 0) & \text{if } e \in U_\mathcal{C} \end{cases}$$

and extend this to U^* as $\kappa_{\widehat{r}}(e \cdot w) = \kappa_{\widehat{r}}(e) \cdot \kappa_{\widehat{r}'}(w)$ where $\widehat{r}'$ is such that $\kappa_{\widehat{r}}(e) = (\widehat{r}, e, \widehat{r}')$. Then:

Lemma 37 *For any $\widehat{r}$ and $\widehat{r}'$, we have $\kappa_{\widehat{r}}(\mathcal{L}(\mathcal{B})^{\widehat{r}'}_{\widehat{r}}) = \mathcal{L}(\mathcal{B}^{\widehat{r}'}_{\widehat{r}})$.*

Proof Pick a finite word $\bar{w} = (\widehat{r}_k, e_k, \widehat{r}'_k)_{0 \leq k \leq n}$ in $(\widehat{\mathbb{R}}_{\geq 0} \times U \times \widehat{\mathbb{R}}_{\geq 0})^*$ accepted by $\mathcal{B}^{\widehat{r}'}_{\widehat{r}}$. By construction of $\mathcal{B}^{\widehat{r}'}_{\widehat{r}}$, it holds $\widehat{r}_0 = \widehat{r}$, and $\widehat{r}'_k = \kappa_{\widehat{r}_k}(e_k)$ for all k. Hence $\bar{w} \in \kappa_{\widehat{r}}(\mathcal{L}(\mathcal{B})^{\widehat{r}'}_{\widehat{r}})$.

Conversely, pick a word $w = (e_k)_{0 \leq k \leq n}$ in $\mathcal{L}(\mathcal{B})^{\widehat{r}'}_{\widehat{r}}$, and consider the word $\bar{w} = \kappa_{\widehat{r}}(w) = (\widehat{r}_k, e_k, \widehat{r}'_k)_{0 \leq k \leq n}$. Again by construction of $\mathcal{B}^{\widehat{r}'}_{\widehat{r}}$, any accepting run of $\mathcal{B}$ on w can be transformed into an accepting run of $\mathcal{B}^{\widehat{r}'}_{\widehat{r}}$ on $\bar{w}$, which entails our result. □

Writing $\pi_U \colon \widehat{\mathbb{R}}_{\geq 0} \times U \times \widehat{\mathbb{R}}_{\geq 0} \to U$ for the projection on the second element of this alphabet (and extending it to sequences in the natural way), we get $\pi_U(\mathcal{L}(\mathcal{B}^{\widehat{r}'}_{\widehat{r}})) = \mathcal{L}(\mathcal{B})^{\widehat{r}'}_{\widehat{r}}$.

Defining $\theta_1(M_{(l,E,\widehat{r})}, [\mathcal{L}(l,l')]^{\widehat{r}'}_{\widehat{r}})$ as a regular union of intervals. We now focus on $\theta_1(M_{(l,E,\widehat{r})}, [\mathcal{L}(l,l')]^{\widehat{r}'}_{\widehat{r}})$: this function maps l' to $\theta_1((E,\widehat{r}), [\mathcal{L}(l,l')]^{\widehat{r}'}_{\widehat{r}})$ (which we write η in the sequel for the sake of readability), and any other state to the empty timed set. To define η as a regular timed set, we progress in three steps: first, we decompose $[\mathcal{L}(l,l')]^{\widehat{r}'}_{\widehat{r}}$ according to the presence of a reset in each sequence of transitions, then we quickly conclude for non-resetting sequences. Finally we discuss the resetting ones.

– **Separating resetting and non-resetting transitions.** For any $\widehat{g} \in \mathbb{J}_{\widehat{r}} = \{\uparrow 0, \widehat{r}\} \cup \widehat{\underbrace{G_{id}}}$ and any $\underbrace{g'} \in \underbrace{G_{id}}$, we define

$$\begin{aligned} W^{id}_{\widehat{g},\underbrace{g'}} = \{\bar{w} = (\widehat{g}_1, e_1, \widehat{g}'_1) \cdots (\widehat{g}_n, e_n, \widehat{g}'_n) \mid \\ \widehat{g}'_n = \widehat{g} \text{ and } \min_{1 \leq k \leq n} \underbrace{e_k} = \underbrace{g'} \\ \text{and } e_k \in U_\varnothing \text{ for all } 1 \leq k \leq n\}. \end{aligned}$$

We decompose $[\mathcal{L}(l,l')]^{\widehat{r}'}_{\widehat{r}}$ as the disjoint union of

$$\bigcup_{\underbrace{g'} \in \underbrace{G_{id}}} [\mathcal{L}(l,l')]^{\widehat{r}'}_{\widehat{r}} \cap \pi_U(W^{id}_{\widehat{r}',\underbrace{g'}})$$

(all runs in $[\mathcal{L}(l,l')]^{\widehat{r}'}_{\widehat{r}}$ that do not contain any resetting transition) and

$$\bigcup_{\underbrace{g'} \in \underbrace{G_{id}}} \bigcup_{(\widehat{g},e) \in \mathbb{J}_{\widehat{r}} \times U_\mathcal{C}} [\mathcal{L}(l,l')]^{\widehat{r}'}_{\widehat{r}} \cap \pi_U\left(W^{id}_{\widehat{g},\underbrace{g'}} \times \{(\widehat{g}, e, \uparrow 0)\} \times (\mathbb{J}_{\widehat{r}} \times U \times \mathbb{J}_{\widehat{r}})^*\right)$$

where the right-hand side of the intersection contains (the projection of) all paths containing at least

one resetting transition labelled $(\widehat{g}, e, \uparrow\!0)$. We write Z for the set $W^{id}_{\widehat{g},\underbrace{g'}} \times \{(\widehat{g}, e, \uparrow\!0)\} \times (\mathbb{J}_{\widehat{r}} \times U \times \mathbb{J}_{\widehat{r}})^*$. Using the decomposition above, we get

$$\eta = \Big(\bigcup_{\underbrace{g'} \in \underbrace{G_{id}}} \bigcup_{w \in [\mathcal{L}(l,l')]^{\widehat{r}'}_{\widehat{r}} \cap \pi_U(W^{id}_{\widehat{r}',\underbrace{g'}})} \theta_1((E,\widehat{r}),w) \Big) \cup \Big(\bigcup_{\underbrace{g'} \in \underbrace{G_{id}}} \bigcup_{(\widehat{g},e) \in \mathbb{J}_{\widehat{r}} \times U_{\mathcal{C}}} \bigcup_{w \in [\mathcal{L}(l,l')]^{\widehat{r}'}_{\widehat{r}} \cap \pi_U(Z)} \theta_1((E,\widehat{r}),w) \Big).$$

– **Conclusion for non-resetting transitions.** In the first part, any path $w \in [\mathcal{L}(l,l')]^{\widehat{r}'}_{\widehat{r}} \cap \pi_U(W^{id}_{\widehat{r}',\underbrace{g'}})$ contains only non-resetting transitions, so that we have $\theta_1((E,\widehat{r}),w) = E \cap \underbrace{g'}$. Hence the first part is a finite union of regular intervals (because $M_{(l,E,\widehat{r})}$ is a regular timed marking). This can be represented by a regular union of intervals of the form $(I, \varnothing, 0, 0)$.

– **Discussion on the non-resetting transitions.** The non-resetting transitions are handled in the following way. Lemma 35 and the regularity of $\mathcal{L}(l,l')^{\widehat{r}'}_{\widehat{r}}$, that we obtained by constructing a finite timed automaton recognising it, are used to reformulate the expression enough to use Parikh's theorem. Then, using the new form arising from the theorem, a final discussion is conducted to separate the case where the closure uses a finite union of sets, and the case where there is an infinite (but regular) union.
For the second part of η, *i.e.*

$$\Big(\bigcup_{\underbrace{g'} \in \underbrace{G_{id}}} \bigcup_{(\widehat{g},e) \in \mathbb{J}_{\widehat{r}} \times U_{\mathcal{C}}} \bigcup_{w \in [\mathcal{L}(l,l')]^{\widehat{r}'}_{\widehat{r}} \cap \pi_U(Z)} \theta_1((E,\widehat{r}),w) \Big)$$

decomposing $w \in \pi_U(Z)$ as $u \cdot e \cdot v$, we have

$$\theta_1((E,\widehat{r}),w) = \theta_1(\theta(\theta((E,\widehat{r}),u),e),v).$$

Since u only contains non-resetting transitions, we have $\theta_1((E,\widehat{r}),u) = E \cap \underbrace{g'}$, and since $u \in W^{id}_{\widehat{g},\underbrace{g'}}$ we have $\theta_2((E,\widehat{r}),u) = \widehat{g}$. Then, since e is a resetting transition, we get

$$\theta_1((E,\widehat{r}),w) = \theta_1(((E \cap \underbrace{g'}) \ltimes \Phi(\widehat{g},e), \uparrow\!0), v).$$

We then have

$$\bigcup_{w \in [\mathcal{L}(l,l')]^{\widehat{r}'}_{\widehat{r}} \cap \pi_U(Z)} \theta_1((E,\widehat{r}),w) = \theta_1\Big(((E \cap \underbrace{g'}) \ltimes \Phi(\widehat{g},e), \uparrow\!0), \mathcal{Q}\Big)$$

where $\mathcal{Q} = \pi_U(W^{id}_{\widehat{g},\underbrace{g'}} \times \{(\widehat{g}, e, \uparrow\!0)\}) \backslash [\mathcal{L}(l,l')]^{\widehat{r}'}_{\widehat{r}}$ is the left-quotient of $[\mathcal{L}(l,l')]^{\widehat{r}'}_{\widehat{r}}$ by $\pi_U(W^{id}_{\widehat{g},\underbrace{g'}} \times \{(\widehat{g}, e, \uparrow\!0)\})$, *i.e.* the set of all finite words $\beta \in U^*$ for which there exists a finite word α in $\pi_U(W^{id}_{\widehat{g},\underbrace{g'}} \times \{(\widehat{g}, e, \uparrow\!0)\})$ such that $\alpha \cdot \beta$ is in $[\mathcal{L}(l,l')]^{\widehat{r}'}_{\widehat{r}}$. Hence it remains to prove that for all $\underbrace{g'} \in \underbrace{G_{id}}$ and all $(\widehat{g}, e) \in \mathbb{J}_{\widehat{r}} \times U_{\mathcal{C}}$, the set

$$\eta' = \theta_1\Big(((E \cap \underbrace{g'}) \ltimes \Phi(\widehat{g},e), \uparrow\!0), \mathcal{Q}\Big),$$

is a regular union of intervals.
Write $E_{e,\widehat{g},\underbrace{g'}} = (E \cap \underbrace{g'}) \ltimes \Phi(\widehat{g},e) \subseteq \mathbb{R}_{\leq 0}$. From Lemma 35, we derive, for any $v \in \mathcal{Q}$:

$$\theta_1((E_{e,\widehat{g},\underbrace{g'}}, \uparrow\!0), v) = E_{e,\widehat{g},\underbrace{g'}} - \sum_{J \in \mathbb{J}_{\uparrow 0} \times U_{\mathcal{C}}} \mathcal{J}_{\uparrow 0}(v)(J) \times \Phi(J).$$

Hence

$$\eta' = \bigcup_{v \in \mathcal{Q}} E_{e,\widehat{g},\underbrace{g'}} - \sum_{J \in \mathbb{J}_{\uparrow 0} \times U_{\mathcal{C}}} \mathcal{J}_{\uparrow 0}(v)(J) \times \Phi(J)$$
$$= E_{e,\widehat{g},\underbrace{g'}} - \bigcup_{v \in \mathcal{Q}} \sum_{J \in \mathbb{J}_{\uparrow 0} \times U_{\mathcal{C}}} \mathcal{J}_{\uparrow 0}(v)(J) \times \Phi(J)$$

For any $v \in \mathcal{Q}$, we write $\mathbf{p}(v)$ for the Parikh vector of $\kappa_{\uparrow 0}(v)$, *i.e.* the function mapping each letter x of $\mathbb{J}_{\uparrow 0} \times U \times \mathbb{J}_{\uparrow 0}$ to the number of occurrences of x along $\kappa_{\uparrow 0}(v)$. We write $\mathbf{p}(\mathcal{Q})$ for the set of all such vectors. Then for all $(\widehat{n}, e) \in \mathbb{J}_{\uparrow 0} \times U_{\mathcal{C}}$, we have $\mathcal{J}_{\uparrow 0}(v)(\widehat{n}, e) = \sum_{\widehat{n}' \in \mathbb{J}_{\uparrow 0}} \mathbf{p}(v)(\widehat{n}, e, \widehat{n}')$. It follows:

$$\eta' = E_{e,\widehat{g},\underbrace{g'}} - \bigcup_{P \in \mathbf{p}(\mathcal{Q})} \sum_{\substack{(\widehat{n},e) \in \mathbb{J}_{\uparrow 0} \times U_{\mathcal{C}} \\ \widehat{n}' \in \mathbb{J}_{\uparrow 0}}} P(\widehat{n}, e, \widehat{n}') \times \Phi(\widehat{n}, e).$$

Now, the set $\mathcal{Q} = \pi_U(W^{id}_{\widehat{g},\underbrace{g'}} \times \{(\widehat{g}, e, \uparrow\!0)\}) \backslash [\mathcal{L}(l,l')]^{\widehat{r}'}_{\widehat{r}}$ is regular, so that by Parikh's theorem, $\mathbf{p}(Q)$ is a semi-linear set; it can be written $\mathbf{p}(Q) = \bigcup_{k=1}^{c} (\mathbf{p}_0^k + \sum_{k'=1}^{d_k} \mathbf{p}_{k'}^k \times \mathbb{N})$, where $\mathbf{p}_{k'}^k \in \mathbb{N}^{\mathbb{J}_{\uparrow 0} \times U_{\mathcal{C}} \times \mathbb{J}_{\uparrow 0}}$ for all $1 \leq k \leq c$ and $0 \leq k' \leq d_k$. Then:

$$\eta' = E_{e,\widehat{g},\underbrace{g'}} - \bigcup_{k=1}^{c} \Big(\sum_{\substack{(\widehat{n},e) \in \mathbb{J}_{\uparrow 0} \times U_{\mathcal{C}} \\ \widehat{n}' \in \mathbb{J}_{\uparrow 0}}} \mathbf{p}_0^k(\widehat{n}, e, \widehat{n}') \times \Phi(\widehat{n}, e) + \bigcup_{(\gamma_{k'})_{k'} \in \mathbb{N}^{d_k}} \sum_{k'=1}^{d_k} \sum_{\substack{(\widehat{n},e) \in \mathbb{J}_{\uparrow 0} \times U_{\mathcal{C}} \\ \widehat{n}' \in \mathbb{J}_{\uparrow 0}}} \gamma_{k'} \cdot \mathbf{p}_{k'}^k(\widehat{n}, e, \widehat{n}') \times \Phi(\widehat{n}, e) \Big).$$

We write $\Gamma_{e,\widehat{g},\underbrace{g'}}$ for the second term, so that $\eta' = E_{e,\widehat{g},\underbrace{g'}} - \Gamma_{e,\widehat{g},\underbrace{g'}}$. For each $1 \leq k \leq c$ and $0 \leq k' \leq d_k$, we define the interval

$$I_{k'}^k = \sum_{(\widehat{n},e) \in \mathbb{J}_{\uparrow 0} \times U_{\mathcal{C}}} \sum_{\widehat{n}' \in \mathbb{J}_{\uparrow 0}} \mathbf{p}_{k'}^k(\widehat{n}, e, \widehat{n}') \times \Phi(\widehat{n}, e),$$

so that

$$\Gamma_{e,\widehat{g},\underline{g}'} = \bigcup_{k=1}^{c} \Big(I_0^k + \bigcup_{(\gamma_{k'})_{k'} \in \mathbb{N}^{d_k}} \sum_{k'=1}^{d_k} \gamma_{k'} \cdot I_{k'}^k \Big).$$

Notice that $I_{k'}^k \subseteq \mathbb{R}_{\geq 0}$ for all k' and k. We then consider two cases:

- first assume that for some k_0 and $k_0' \geq 1$ and some $(\widehat{n}_0, e_0) \in \mathbb{J}_{\uparrow 0} \times U_{\mathcal{C}}$, it holds $\mathbf{p}_{k_0'}^{k_0}(\widehat{n}_0, e_0, \uparrow 0) > 0$ and $\Phi(\widehat{n}_0, e_0)$ has positive length. Consider the interval $L_\gamma = I_0^{k_0} + \gamma \cdot I_{k_0'}^{k_0}$ for any $\gamma \in \mathbb{N}$. Then clearly $L_\gamma \subseteq \Gamma_{e,\widehat{g},\underline{g}'}$ for all γ. Moreover, by definition of k_0 and k_0', the length of $I_{k_0'}^{k_0}$ is positive, so that the length of $\gamma \cdot I_{k_0'}^{k_0}$ tends to infinity with γ. It follows that for some $\alpha \in \mathbb{Q}_{\geq 0}$, $\uparrow \alpha \subseteq \bigcup_{\gamma \in \mathbb{N}} L_\gamma \subseteq \Gamma_{e,\widehat{g},\underline{g}'}$. Then $\Gamma_{e,\widehat{g},\underline{g}'} \cap \downarrow \alpha$ has finite granularity, so that it is a finite union of intervals of the form $I_{k'}^k \cap \downarrow \alpha$. It follows that $\Gamma_{e,\widehat{g},\underline{g}'}$ is a finite union of intervals, and can hence be represented by a regular union of intervals.
  Now, by hypothesis, E can be represented as a regular union of intervals. By Prop. 22, so can $E_{e,\widehat{g},\underline{g}'}$. Then by Prop. 7, $\eta' = E_{e,\widehat{g},\underline{g}'} - \Gamma_{e,\widehat{g},\underline{g}'}$ can be represented as a regular union of intervals.
- We now assume that for all k and $k' \geq 1$ and all $(\widehat{n}, e) \in \mathbb{J}_{\widehat{\tau}} \times U_{\mathcal{C}}$, either $\mathbf{p}_{k'}^k(\widehat{n}, e, \uparrow 0) = 0$, or $\Phi(\widehat{n}, e)$ has length 0. Then for all $1 \leq k \leq n$ and $1 \leq k' \leq d_k$, $I_{k'}^k$ contains a single element, which is a rational. Then $-\Gamma_{e,\widehat{g},\underline{g}'} = \{-\gamma \mid \gamma \in \Gamma_{e,\widehat{g},\underline{g}'}\}$ can be represented as a regular union of intervals. By Prop. 7, we get that η' can also be represented by a regular union of intervals in that case.

In the end, η is the union of two sets that can be represented as regular unions of intervals, thus by Prop. 7, it can also be represented as a regular union of intervals. □

5 Experimentations

In order to evaluate the possible improvement of our approach compared to the technique proposed in [27], we implemented and compared the performances of both approaches. Sources can be downloaded at `http://people.irisa.fr/Nicolas.Markey/download/DOTA.zip`.

5.1 Comparison of the approaches

In the approach of [27], the set of actual configurations is stored as a marking. If an observable action a occurs after some delay d, the algorithm computes the set of all possible configurations reached after delay d (possibly following silent transitions), and applies from the resulting markings the set of all available transitions labelled a. This amounts to computing the functions $\mathbf{O}_d$ and $\mathbf{O}_a$ at each observation. There is also a timeout, which makes the algorithm update the marking (with $\mathbf{O}_d$) regularly if no action is observed. The computation of $\mathbf{O}_d$ is heavily used, and has to be performed very efficiently for this technique to be usable at runtime.

In our approach, we use *regular timed markings* to store sets of possible configurations. Given a timed marking, when an action a is observed after some delay d, we can easily compute the set of configurations reachable after delay d, and have to apply $\mathbf{O}_a$ and recompute the τ-closure. Following [10], $\mathbf{O}_a$ can be performed as a series of set operations on intervals. The τ-closure can be performed as a series of subtractions between an interval and regular unions of intervals (see Lemma 35). Those regular unions can be precomputed; while this may require exponential time and space to compute and store, this makes the simulation of delay transitions at runtime very efficient.

5.2 Implementation

In our experimentations, in order to only evaluate the benefits of the precomputation and of the use of τ-closures in our approach compared to that of [27], we use the same data structure for both algorithms. The only difference lies in the functions computing the observable and the delay transitions. As a consequence, both implementations benefit from the data structure we chose for representing timed intervals, which allows us to compute basic operations in linear time. Also, both structures use the same reachability graphs for either computing the sets of reachable configurations or the Parikh images.

Our implementation is written in Python3. One-clock timed automata and both state-evaluation algorithms are instances of an abstract class of *automata over timed markings* [10]; timed markings are implemented in a library `TILib`. Simulations of those automata are performed using an object called `ATDRunner`, which takes an automaton over timed markings and simulates its transitions according to the actions it observes on a given *input channel*. It may also write what it does on an *output channel*. A channel is basically a way of communicating with `ATDRunner`s.

In order to evaluate, at runtime, the set of configurations of a given one-clock timed automaton with silent transitions, stored in an object `OTAutomata`, we first generate a diagnoser object, either a `DiagOTA` or

	#locations	#silent trans.	#observ. trans	precomp. time	actions DiagOTA	actions TripakisDOTA	ratio actions	delay DiagOTA	delay TripakisDOTA	ratio delays
Example1	3	6	14	124.7s.	0.011s.	0.011s.	1.02	0.018ms.	0.024s.	7.50×10^{-4}
Example2	3	6	12	2.8s.	0.007s.	0.005s.	1.40	0.017ms.	0.017s.	9.97×10^{-4}
Example3	4	6	14	0.3s.	0.026s.	0.036s.	0.71	0.019ms.	0.024s.	7.97×10^{-4}
Example4	4	7	21	599.1s.	0.019s.	0.023s.	0.83	0.017ms.	0.023s.	5.97×10^{-4}
Example5	7	5	46	4.9s.	0.056s.	0.024s.	2.31	0.012ms.	0.021s.	5.73×10^{-4}
Example6	7	5	43	4.1s.	0.070s.	0.020s.	3.37	0.011ms.	0.017s.	6.28×10^{-4}
Example7	7	10	49	6.6s.	0.073s.	0.043s.	1.69	0.014ms.	0.070s.	1.99×10^{-4}
Example8	7	10	51	7.6s.	0.057s.	0.125s.	0.46	0.019ms.	0.165s.	1.13×10^{-4}
Example9	7	10	54	7.0s.	0.076s.	0.077s.	0.99	0.010ms.	0.112s.	0.92×10^{-4}

Table 1 Benchmarks for 9 random examples over 20 runs with 20 actions each. This table shows the average computation time for updating the timed marking of reachable configurations when an action or a delay is observed. It can be seen that for observable transitions, the computation times are comparable; on the other hand, for delay transitions, our approach outperforms the approach of Tripakis by several orders of magnitude.

a `TripakisDOTA`, depending of which version we want to use. Then we launch two threads: one is an `ATDRunner` simulating the timed automaton, listening to some channel object `Input`, and writing every observable action it performs on some other channel object `Comm`. The other one is another `ATDRunner` simulating the diagnoser and listening to the `Comm` channel.

In a `DiagOTA` object, which corresponds to our approach, we have already precomputed the relevant timed intervals; observable transitions are then made by operations over timed markings. In such a simulation, we can thus keep track of which configurations may have been reached, but also predict which configurations may be reached in the future and the exact time before we can reach them.

In a `TripakisDOTA` object, which corresponds to the approach of [27], observable and delay transitions are simulated by computing all configurations reachable through that observable action or delay, also allowing arbitrarily many silent transitions. This does not allow for prediction.

5.3 Results

Table 1 reports on the performances of both implementations on a small set of (randomly generated) examples. Those examples are distributed with our prototype. In Table 1, we give the important characteristics of each automaton (number of locations and of silent transitions), the amount of precomputation time used by our approach, and the average time (over 20 random runs) used in the two approaches to simulate observable and delay transitions.

As could be expected, our approach outperforms the approach of [27] on delay transitions by several orders of magnitude in all cases. The performances of both approaches are comparable when simulating observable transitions.

The precomputation phase of our approach is intrinsically very expensive. In our examples, it takes from less than a second to 10 minutes, and it remains to be understood which factors make this precomputation phase more or less difficult. We may also refine our implementation of the computation of Parikh images, which is heavily used in the precomputation phase.

6 Towards efficient diagnosis for n-clocks timed automata

In this section, we extend (part) of our formalism to n-clock timed automata. Precisely, we define a generalized gauge operator, and prove that it can be used to compute a θ function. As for one-clock automata, we show that it is enough to consider *linear* timed markings, that θ handles.

However we do not propose an extension of regular unions of intervals and thus do not prove that we can furthermore restrict the study to regular timed markings.

We first extend the basic definitions in Section 6.1, and then use them to define a generalized $\ltimes$-operator and prove that it can be used to compute τ-closures for any number of clocks (Section 6.2).

6.1 Timed automata

In the n-clock setting, a timed automaton is equipped with its set of clocks, and transitions can reset any number of those. To generalize the guards on the automaton transitions, we use functions associating an interval of $\mathcal{I}_{\mathbb{Q}_{\geq 0}}$ with each clock.

Definition 38 A *timed automaton* over Σ is a tuple $\mathcal{A} = (L, \{l_0\}, \mathcal{C}, \Delta, \mathcal{F})$, where L is a finite set of locations, $l_0 \in L$ is the initial location, $\mathcal{C}$ is a set of variables called clocks, $\Delta \subseteq L \times \mathcal{I}_{\mathbb{Q}_{\geq 0}}^{\mathcal{C}} \times (\Sigma \uplus \{\tau\}) \times 2^{\mathcal{C}} \times L$ is the set of transitions, and $\mathcal{F} \subseteq L$ is a set of final locations.

A clock valuation is usually defined as a function $v \in \mathbb{R}_{\geq 0}^{\mathcal{C}}$. As we use negative values to store *future* clock values, we will sometimes consider functions to $\mathbb{R}$ instead. To simplify furthermore the technical discussions, we fix an (arbitrary) order on the clocks, and will hence allow ourselves to consider valuations as vectors in $\mathbb{R}^n$. Using this, we will generalize the notions introduced in 2.1 for intervals to intersections of constraints on different dimensions. For a subset $\mathbb{K}$ of $\mathbb{R}$, we consider $\widehat{r} \in \widehat{\mathbb{K}}_{\geq 0}^n$ and $\underset{\sim}{r} \in \mathbb{K}_{\geq 0}^n$ [3]. We define the sum of a subset E of $\mathbb{R}^n$ with an integer $d \in \mathbb{R}$ as the translation of the set on all dimensions: $E + d = \{(e_1 + d, .., e_n + d) \mid (e_1, .., e_n) \in E\}$. This corresponds to d time units elapsing. Using these definitions, we can extend atomic, finite and linear *timed sets* to n-clocks simply by considering sets and constraints in n dimensions. We note $\mathcal{T}(\mathbb{R}^n)$ the set of linear timed sets of $\mathbb{R}^n$. A *clock reset* over a subset of clocks $X \subseteq \mathcal{C}$, denoted $v_{[X \leftarrow 0]}$, is an operation that projects all clocks in X to 0, while leaving the other ones unchanged. This definition can be trivially extended to sets.

To discuss timed predecessors and successors of a set of clock valuations we note respectively $\mathsf{Pre}(E) = \bigcup_{d \in \mathbb{R}_{\geq 0}} E - d$ and $\mathsf{Post}(E) = \bigcup_{d \in \mathbb{R}_{\geq 0}} E + d$.

A configuration of a timed automaton is a pair $(l, v) \in L \times \mathbb{R}_{\geq 0}^n$. The semantics can still be defined as an infinite transition system with delay transitions $(l, v) \xrightarrow{\delta} (l, v')$ where $\delta \in \mathbb{R}_{\geq 0}$ and $v' = v + \delta$, and action transitions $(l, v) \rightarrow_e (l', v')$ for any $e = (l, G, a, X, l')$ in Δ and $v \in G$ with $v' = v_{[X \leftarrow 0]}$.

With this semantics, the notion of sequences and its associated notations can be directly reused. We extend markings to elements of $\mathbb{M} = (\mathcal{P}(\mathbb{R}_{\geq 0}^n))^L$ and use similar definitions for $\mathbf{O}_a$ for an observable action $a \in \Sigma$:

$$\mathbf{O}_a(m) \colon l' \mapsto \{v' \in \mathbb{R}_{\geq 0}^n \mid \exists l \in L.\ \exists v \in m(l).\ (l, v) \overset{0}{\rightsquigarrow}_a (l', v')\}$$

and $\mathbf{O}_d$:

$$\mathbf{O}_d(m) \colon l' \mapsto \{v' \in \mathbb{R}_{\geq 0}^n \mid \exists l \in L.\ \exists v \in m(l).\ (l, v) \overset{d}{\rightsquigarrow}_\varepsilon (l', v')\}.$$

6.2 τ-closures

Using the previous definitions, we can extend timed markings to n dimensions and get back every definition and result of Section 4.1 by just considering valuations in $\mathbb{R}_{\geq 0}^n$ instead of $\mathbb{R}_{\geq 0}$. In this context, $\uparrow 0$ has to be read as the intersection of the corresponding one dimensional constraints on all dimensions. We hence focus here on how to compute the τ-closures using linear timed markings, as done in Section 4.2. For this, we first define and formalize the operator transforming a set of (potential) configurations by the effect of a guard and a set of clock resets.

Definition 39 For two sets E and G (a guard) and $X' \subseteq \mathcal{C}$, we let

$$E \ltimes_{X'} G = \{v \in \mathbb{R}^n \mid \exists d \geq 0,\ v \in ((E+d) \cap G)_{[X' \leftarrow 0]} - d\} \tag{1}$$

Example 40 Consider a transition with guard $G = 1 \leq x \leq 2$ in a timed automaton over the two clocks $\mathcal{C} = \{x, y\}$. We take the set of potential valuations $E = -1 \leq x \leq 0 \wedge y \geq 0 \wedge y - x \leq 1$. Both E and G are depicted in Fig 16, together with $E \ltimes_{\{x\}} G$ and $E \ltimes_{\{y\}} G$.

Notice how by adding delays $d \in \mathbb{R}$ to these sets we can see that the gauge corresponds to the resets of clocks for valuations in $E + d \cap G$. ◁

This operator generalizes the operator we defined for the one-clock case: indeed, for $\mathcal{C} = \{x\}$:

$$\begin{aligned} E \ltimes_{\mathcal{C}} G &= \{v \in \mathbb{R} \mid \exists d \geq 0,\ v \in ((E + d) \cap G)_{[x \leftarrow 0]} - d\} \\ &= \{-d \mid d \in \mathbb{R}_{\geq 0} \mid ((E + d) \cap G) \neq \varnothing\} \end{aligned}$$

which is equal to $E \ltimes G$ (the one-clock operator) by Prop. 20.

Thanks to this operator, we can define the effect of a transition on an atomic timed set:

[3] Both open and closed intervals may appear in the same element on different dimensions

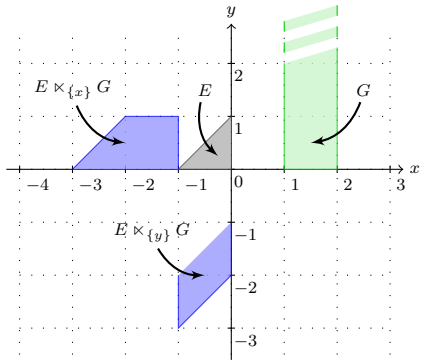

Fig. 16 Effect of the gauge on individual clocks for $E = -1 \leq x \leq 0 \wedge y \geq 0 \wedge y - x \leq 1$ and $G = 1 \leq x \leq 2$.

Definition 41 For an atomic timed set $T = (E, \widehat{r})$ of $\mathcal{T}(\mathbb{R}^n)$ and a transition $e = (l, G = \widehat{e} \cap \underline{e}, \tau, X', l')$ of U:

$$\theta((E,\widehat{r});e) = \begin{cases} (\varnothing; \uparrow 0) & \text{if } \mathsf{Post}(E) \cap \widehat{r} \cap G = \varnothing \\ (E \cap \mathsf{Pre}(\widehat{r} \cap G); \widehat{r} \cap \widehat{e}) & \\ \quad \text{if } \mathsf{Post}(E) \cap \widehat{r} \cap G \neq \varnothing \text{ and } X' = \varnothing & \\ (E \ltimes_{X'} (\widehat{r} \cap G); (\widehat{r} \cap \widehat{e})_{[X' \leftarrow \uparrow 0]}) & \\ \quad \text{if } \mathsf{Post}(E) \cap \widehat{r} \cap G \neq \varnothing \text{ and } X' \neq \varnothing. & \end{cases}$$

Notice that $E \cap \mathsf{Pre}(\widehat{r} \cap G) = E \ltimes_{\varnothing} (\widehat{r} \cap G)$, so that the n-clock operator can indeed represent the effect of any transition in a timed automaton. Although these cases can be grouped together, we keep them separate as the actual computation needed when there is no reset is much less involved. The definition of θ is illustrated in Fig. 17 for a two-clocks automaton.

Definition 41 (cont.) As in the one-clock case, we extend θ to sequences of transitions inductively by letting $\theta((E;\widehat{r}),\varepsilon) = (E;\widehat{r})$ and, for $w \in U^+$,

$$\theta((E;\widehat{r}), w{\cdot}e) = \begin{cases} \theta(\theta((E;\widehat{r}),w),e) & \text{if } \mathsf{tgt}(w) = \mathsf{src}(e) \\ (\varnothing; \uparrow 0) & \text{otherwise.} \end{cases}$$

Finally we extend this definition to linear timed sets by letting

$$\theta(\{T_k \mid k \in K\}, w) = \{\theta(T_k, w) \mid k \in K\}.$$

Using these definitions, we can prove that θ corresponds to the effect of a sequence of transitions from a timed set. The following lemma extends Lemma 26. Notice that the proof has to be adapted, not only to the new gauge definition but also to the diagonal constraints arising in timed automata with multiple clocks.

Lemma 42 *Let T be a linear timed set and $w \in U^*$. Then for any $d \in \mathbb{R}_{\geq 0}$ and any $v \in \mathbb{R}^n_{\geq 0}$,*

$$v \in f_{\theta(T,w)}(d) \;\Leftrightarrow\; \exists d_0 \in [0;d].\ \exists v' \in f_T(d_0).\ (\mathsf{src}(w), v') \xrightarrow{d-d_0}_w (\mathsf{tgt}(w), v).$$

Proof We carry the proof for the case where $T = (E;\widehat{r})$ is an atomic timed set. The extension to unions of atomic timed sets is straightforward. We begin with the case where w is a single transition $e = (l, G = \widehat{e} \cap \underline{e}, \tau, X', l')$. In case E is empty, then also $f_{\theta(T,e)}(d)$ is empty, and $\forall d_0, f_T(d_0) = \varnothing$, so the result holds. We now assume that E is not empty, and consider three cases:

- if $\mathsf{Post}(E) \cap \widehat{r} \cap G = \varnothing$, then $\theta(T,e) = (\varnothing; \uparrow 0)$. On the other hand, for any d_0 and any $v' \in f_T(d_0)$, it holds $v' \in \mathsf{Post}(E) \cap \widehat{r}$, so that $v' \notin G$, and the transition cannot be taken from that valuation. Hence both sides of the equivalence evaluate to false, and the equivalence holds.
- now assume that $\mathsf{Post}(E) \cap \widehat{r} \cap G \neq \varnothing$, and consider the case where X' is empty. We proceed by double inclusion. We first take a valuation $v \in f_{\theta(T,w)}(d)$ and exhibit a predecessor of this configuration in $f_T(d_0) \cap G$ for a d_0 to be constructed.
  $v \in f_{\theta(T,w)}(d)$ means that $v \in (E \cap \mathsf{Pre}(\widehat{r} \cap G)) + d \cap \widehat{r} \cap \widehat{e}$. Equivalently, $v \in (E + d) \cap \mathsf{Pre}(\widehat{r} + d \cap G + d) \cap \widehat{r} \cap \widehat{e}$. As $\widehat{e}$ encodes the lower bounds on individual clocks of G and $\mathsf{Pre}(\widehat{r} + d \cap G + d)$ keeps all diagonal constraints of $\widehat{r} \cap G$, we have that $v \in (E + d) \cap \mathsf{Pre}(\widehat{r} + d \cap G + d) \cap \mathsf{Post}(\widehat{r} \cap G)$. Hence $\exists d' \in [0,d],\ v - d' \in E + d - d' \cap \widehat{r} \cap G$. By taking $d_0 = d - d'$ and $v' = v - d + d_0 = v - d'$ we can write $\exists d_0 \in [0,d], \exists v' \in f_T(d_0) \cap \widehat{r} \cap G$. The first inclusion then holds by definition of w.
  Conversely, if $d_0 \in [0;d]$ and $v' \in f_T(d_0)$ exist such that $(\mathsf{src}(w), v') \xrightarrow{d-d_0}_w (\mathsf{tgt}(w), v)$, then $v' \in E + d_0 \cap \widehat{r}$ and as there is no reset on w, $v = v' + d - d_0$, $v \in E + d \cap \widehat{r} + d - d_0 \subseteq E + d \cap \widehat{r}$. Furthermore, $\exists d_1 \leq d - d_0$ such that $v' + d_1 \in G$. More precisely, $v' + d_1 \in \widehat{r} \cap G$ as $v' \in \widehat{r}$ and this set has no upper bound. As $v = v' + d - d_0$, we have $v \in \mathsf{Pre}(\widehat{r} \cap G) + d$ and $v \in \widehat{e}$ as $d_1 \leq d - d_0$. By aggregating the constraints, we obtain $v \in (E + d) \cap \mathsf{Pre}((\widehat{r} \cap G \cap) + d) \cap \widehat{r} \cap \widehat{e}$ *i.e.* the left-hand side of the equivalence.
- Consider $\mathsf{Post}(E) \cap \widehat{r} \cap G \neq \varnothing$ and $X' \neq \varnothing$. In this case $v \in f_{\theta(T,w)}(d)$ means that $\exists \delta \geq 0,\ v \in (\widehat{r} \cap G \cap (E + \delta))_{[X' \leftarrow 0]} - \delta + d$. Notice that as $v \in \mathbb{R}^n_{\geq 0}$, it comes that $d - \delta \geq 0$. Let $v'_{[X' \leftarrow 0]} = v + \delta - d \in (f_T(\delta) \cap G)_{[X' \leftarrow 0]}$ It comes that we can construct $v' \in f_T(\delta) \cap G$ from $v'_{[X' \leftarrow 0]}$[4]. By taking $d_0 = \delta$ we have that $d_0 \in [0,d]$, $v' \in f_T(d_0)$ such that $v' \xrightarrow{d-d_0}_w v' + d - d_0 = v$.
  Conversely if for a given $d_0 \in [0,d]$ and a given $v' \in f_T(d_0)$, we have that $(\mathsf{src}(w), v') \xrightarrow{d-d_0}_w (\mathsf{tgt}(w), v)$, then there exists $d_1 \in [0, d - d_0]$ such that $v' + d_1 \in G$ and the transition is taken from v'. By definition of f_T, $v' + d_1 \in f_T(d_0 + d_1) \cap G$. We have that

[4] Possibly several such v' exists, we consider one of them.

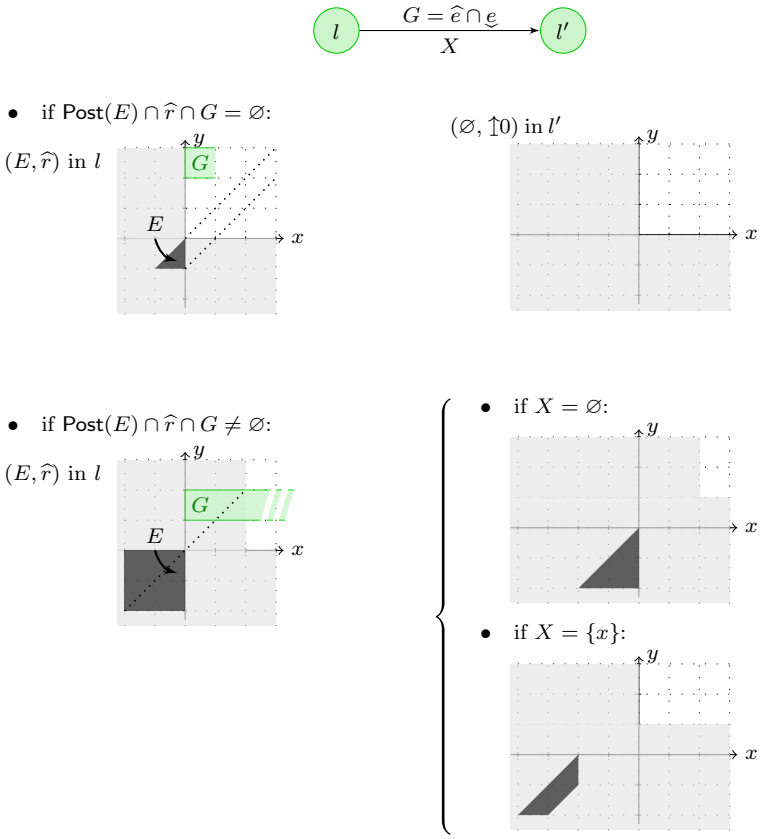

Fig. 17 Representation of the effect of silent transition (l, G, τ, X, l') in three cases.

$v = (v' + d_1)_{[X' \leftarrow 0]} + d - d_0 - d_1 \in (f_T(d_0 + d_1) \cap G)_{[X' \leftarrow 0]} + d - d_0 - d_1$. Hence by taking $\delta = d_0 + d_1$ we ensure that $v \in (E \ltimes_{X'} G) + d$. Furthermore, $d - d_0 - d_1 \geq 0$ as $d_1 \leq d - d_0$ thus $\forall x \in X'$, $v(x) \geq 0$, and $v_{|X'} \in (\widehat{r} \cap \widehat{e})_{|X'}$ as $v_{|X'} \in \mathsf{Post}_{X'}(v' + d_1)$ ($d1 \leq d - d_0$) and $v' + d_1 \in f_T(d_0 + d_1) \cap G \subseteq \widehat{r} \cap G \subseteq \widehat{r} \cap \widehat{e}$. This proves the left part of the equivalence.

We now extend this result to sequences of transitions. The case where $w = \varepsilon$ is straightforward. Now assume that the result holds for some word w, and consider a word $w \cdot e$. In case $\mathsf{tgt}(w) \neq \mathsf{src}(e)$, the result is trivial.

The case of single transitions has been handled just above. We thus consider the case of $w \cdot e$ with $w \in U^+$. First assume that $v' \in f_{\theta(T, w \cdot e)}(d)$, and let $T' = \theta(T, w)$. Then $v' \in f_{\theta(T', e)}(d)$, thus there exist $0 \leq d_0 \leq d$ and $v \in f_{T'}(d_0)$ s.t. $(\mathsf{src}(e), v) \xrightarrow{d - d_0}_e (\mathsf{tgt}(e), v')$. Since $v \in f_{T'}(d_0)$, there must exist $0 \leq d_1 \leq d_0$ and $v'' \in f_T(d_1)$ such that $(\mathsf{src}(w), v'') \xrightarrow{d_0 - d_1}_w (\mathsf{tgt}(w), v)$. This way, we have found a value d_1 with $0 \leq d_1 \leq d$ such that $(\mathsf{src}(w), v'') \xrightarrow{d - d_1}_{w \cdot e} (\mathsf{tgt}(e), v')$.

Conversely, if $(\mathsf{src}(w), v'') \xrightarrow{d - d_1}_{w \cdot e} (\mathsf{tgt}(e), v')$ for some $0 \leq d_1 \leq d$ and $v'' \in f_T(d_1)$, then we have $(\mathsf{src}(w), v'') \xrightarrow{d_0 - d_1}_w (\mathsf{tgt}(w), v) \xrightarrow{d - d_0}_e (\mathsf{tgt}(e), v')$ for some $d_0 \in [d_1; d]$ and some v. We prove $v \in f_{\theta(T,w)}(d_0)$: indeed, we have $d_1 \in [0; d_0]$, and $v'' \in f_T(d_1)$ such that $(\mathsf{src}(w), v'') \xrightarrow{d_0 - d_1}_w (\mathsf{tgt}(w), v)$, which by induction hypothesis entails $v \in f_{\theta(T,w)}(d_0)$. Thus we have $d_0 \in [0; d]$ and $v \in f_{T'}(d_0)$, where $T' = \theta(T, w)$, such that $(\mathsf{tgt}(w), v) \xrightarrow{d - d_0}_e (\mathsf{tgt}(e), v')$, which means $v' \in f_{\theta(T', e)}(d)$, and concludes the proof. □

As in the one-clock case, this semantic characterization of $\theta(F, w)$ entails the following, ensuring that the result of a computation by θ does not depend on the representation.

Corollary 43 *For any sequence w of transitions, and any two equivalent linear timed sets T and T', the linear timed sets $\theta(T, w)$ and $\theta(T', w)$ are equivalent.*

We can once more extend θ to linear timed markings in the same way and generalize Theorem 28 to n clocks. The proof is the same for $v \in \mathbb{R}^n_{\geq 0}$, using Lemma 42.

By considering the extensions of θ to languages, we immediately get:

Corollary 44 *For any linear timed marking M, it holds $\theta(M) \equiv M^\tau$.*

This proves that our θ operator (and the underlying gauge) is a correct representation of the semantic of a

(silent) timed automaton, and that linear timed markings are sufficient to represent the semantics of timed automata.

Example 45 Consider the timed automaton with two clocks depicted in Fig. 18 (the τ labels are omitted). The construction with l_0, l_1 and l_2 allows to obtain the atomic timed set $(x \in [-1,0] \wedge y \in [-1,0]; x, y \geq 0)$ as input for l_3 [5]. The closure of l_3 is depicted on the side with the two guards generating it. It is composed of several infinite behaviours. Notice that there is some regularity in the structure, but that it can be complex: several nested infinite repetitions appear. ◁

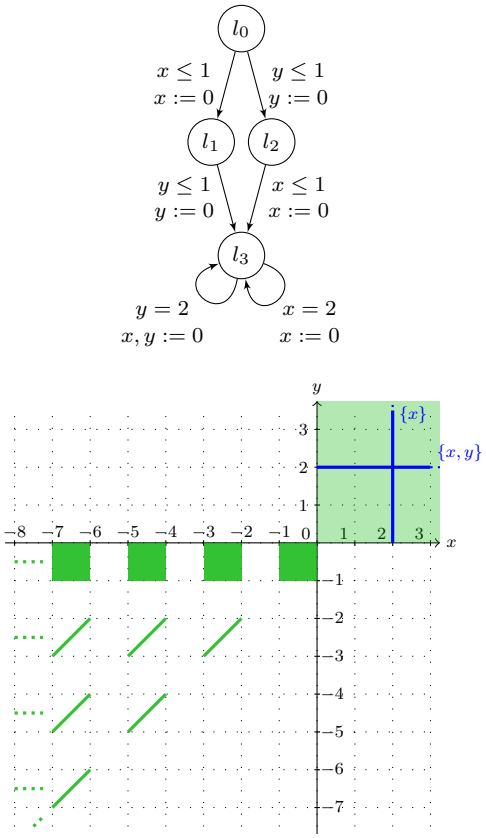

Fig. 18 A simple timed automaton and its associated timed set.

6.3 Stability of a representable class

We do not define regular timed sets and markings for multiple clocks, but in the following, we prove that finite timed markings are enough to represent the closure.

[5] We slightly abuse the construction to take the union of sets for two timed sets sharing the same filter.

Proposition 46 *Consider a timed automaton $\mathcal{A}$ and a regular language $\mathcal{L} \subseteq \mathcal{L}_\Delta(\mathcal{A})$ included in the language of transitions. Given a finite timed marking M, $\theta(M, \mathcal{L})$ is a finite timed marking.*

Proof Let M be a finite timed marking: then M can be written as the finite union of atomic timed markings $M_{(l,E,\widehat{r})}$, defined as $M_{(l,E,\widehat{r})}(l) = f_{(E;\widehat{r})}$ and $M_{(l,E,\widehat{r})}(l') = f_{(\varnothing;\uparrow 0)}$ for all $l' \neq l$.

In the end, any regular timed marking M can be written as the finite union $\bigsqcup_{k\in K} M_k$ of atomic timed markings. Thus we have $\theta(M) \equiv \bigsqcup_{k\in K} \theta(M_k)$ and it is enough to prove the property for atomic timed markings.

We write $\theta_1((E;\widehat{r}), w)$ and $\theta_2((E;\widehat{r}), w)$ for the first and second components of $\theta((E;\widehat{r}), w)$, and use the same generalization to timed markings than in the one-clock case.

Note that:

- $\theta_2((E;\widehat{r}), w) = \widehat{r} \cap \bigcap_{k<n} \widehat{e}_k$ if $w = e_1 \dots e_n \in {U_\varnothing}^*$ is a sequence of consecutive non-resetting transitions.
- $\theta_2((E;\widehat{r}), w \cdot e) = (\theta_2((E;\widehat{r}), w) \cap \widehat{e})_{[X' \leftarrow \uparrow 0]}$ if e is a resetting transition for X'.

We define the set of upper bounds of guards appearing in U as

$$\mathcal{G} = \{\widehat{e} \mid \exists\, e = (l, G = \widehat{e} \wedge \underline{e}, X', \theta, l') \in U\}$$

and using this set $\mathbb{J}_{\widehat{r}} = \{\widehat{g} \in \mathbb{R}^n_{\geq 0} \mid \forall i \in [1,n],\ \exists\, \widehat{e} \in \mathcal{G} \cup \{\widehat{r}, \uparrow 0\}, \widehat{e}_i = \widehat{g}_i\}$. It is easy to see that $\mathbb{J}_{\widehat{r}}$ is a finite set and $\theta_2((E;\widehat{r}), w) \in \mathbb{J}_{\widehat{r}}$. Hence a finite number of atomic timed sets is sufficient to describe the closure of a finite timed marking, *i.e.* the closure is a finite timed set. Indeed, there is a finite number of filters $\widehat{r}$ in the representation of M (one per M_k), no more than $|\mathbb{J}_{\widehat{r}}|$ different atomic timed sets are needed per location of the automaton for each $\widehat{r}$, and there is a finite number of locations. □

7 Conclusion and future works

In this paper, we presented a novel approach to solve the state estimation problem, and in particular to efficiently compute the τ-closure in the case of partially observable one-clock timed automata; it builds on a kind of powerset construction for automata over timed domains, using our new formalism of *linear timed sets* to represent the evolution of the set of reachable configurations of the automaton. We prove that the semantics - and in particular the τ-closure - of a timed automaton can be computed using *regular* timed sets, a finitely

representable subclass of linear timed sets. We furthermore show that the full class of regular timed sets is needed to represent general one-clock timed automata.

We extend the basis of our approach to timed automata with *multiple* clocks, and prove that finite timed sets are again enough to compute the τ-closure.

Our prototype implementation shows the feasibility of our approach on small examples of one-clock automata.

There remains space for improvements in many directions: first, our implementation can probably be made more efficient on the precomputation phase, and at least we need to better understand why some very small examples are so hard to handle.

A natural extension of these results is to introduce a (representable) notion of regularity for multiple clocks automata and to prove that it is large enough to represent the semantics and τ-closure.

This is by no means immediate, as first the presence of multiple clocks allows involved behaviours, and second (and most importantly), the proofs proposed in the one-clock case rely on the structure of the $\ltimes$ operator, and the ability to simply "count" resetting transitions (*i.e.* the order of transitions only slightly matters). This is not true for multiple clocks, where the orders of the resets is central to the dynamics.

Another possible direction of research could target priced timed automata, with the aim of monitoring the cost of the execution in the worst case.

References

1. Rajeev Alur and David L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126(2):183–235, April 1994.
2. Christel Baier, Nathalie Bertrand, Patricia Bouyer, and Thomas Brihaye. When are timed automata determinizable? In Susanne Albers, Alberto Marchetti-Spaccamela, Yossi Matias, Sotiris Nikoletseas, and Wolfgang Thomas, editors, *Automata, Languages and Programming*, pages 43–54, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.
3. 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.
4. Johan Bengtsson and Wang Yi. Timed automata: Semantics, algorithms and tools. In Jörg Desel, Wolfgang Reisig, and Grzegorz Rozenberg, editors, *Lectures on Concurrency and Petri Nets*, volume 2098 of *Lecture Notes in Computer Science*, pages 87–124. Springer-Verlag, 2004.
5. Béatrice Bérard, Paul Gastin, and Antoine Petit. On the power of non-observable actions in timed automata. In Claude Puech and Rüdiger Reischuk, editors, *STACS 96*, pages 255–268, Berlin, Heidelberg, 1996. Springer Berlin Heidelberg.
6. Nathalie Bertrand, Thierry Jéron, Amélie Stainer, and Moez Krichen. Off-line test selection with test purposes for non-deterministic timed automata. *Logical Methods in Computer Science*, 8(4), 2012.
7. Nathalie Bertrand, Thierry Jéron, Amélie Stainer, and Moez Krichen. Off-line test selection with test purposes for non-deterministic timed automata. *Logical Methods in Computer Science*, 8(4), 2012.
8. Nathalie Bertrand, Amélie Stainer, Thierry Jéron, and Moez Krichen. A game approach to determinize timed automata. *Formal Methods in System Design*, 46(1):42–80, February 2015.
9. Patricia Bouyer, Fabrice Chevalier, and Deepak D'Souza. Fault diagnosis using timed automata. In Vladimiro Sassone, editor, *Proceedings of the 8th International Conference on Foundations of Software Science and Computation Structure (FoSSaCS'05)*, volume 3441 of *Lecture Notes in Computer Science*, pages 219–233. Springer-Verlag, April 2005.
10. Patricia Bouyer, Samy Jaziri, and Nicolas Markey. On the determinization of timed systems. In Alessandro Abate and Gilles Geeraerts, editors, *Proceedings of the 15th International Conferences on Formal Modelling and Analysis of Timed Systems (FORMATS'17)*, volume 10419 of *Lecture Notes in Computer Science*, pages 25–41. Springer-Verlag, September 2017.
11. Patricia Bouyer, Samy Jaziri, and Nicolas Markey. Efficient timed diagnosis using automata with timed domains. In Christian Colombo and Martin Leucker, editors, *Runtime Verification - 18th International Conference, RV 2018, Limassol, Cyprus, November 10-13, 2018, Proceedings*, volume 11237 of *Lecture Notes in Computer Science*, pages 205–221. Springer, 2018.
12. Edmund M. Clarke, E. Allen Emerson, and Joseph Sifakis. Model checking: algorithmic verification and debugging. *Communications of the ACM*, 52(11):74–84, November 2009.
13. Edmund M. Clarke, Orna Grumberg, and Doron A. Peled. *Model checking*. MIT Press, 2000.
14. Edmund M. Clarke, Thomas A. Henzinger, Helmut Veith, and Roderick Bloem. *Handbook of Model Checking*. Springer-Verlag, April 2018.
15. Jean-Christophe Filliâtre. Deductive software verification. *International Journal on Software Tools for Technology Transfer*, 13(5):397–403, October 2011.
16. Olivier Finkel. Undecidable problems about timed automata. In Eugene Asarin and Patricia Bouyer, editors, *Proceedings of the 4th International Conferences on Formal Modelling and Analysis of Timed Systems (FORMATS'06)*, volume 4202 of *Lecture Notes in Computer Science*, pages 187–199. Springer-Verlag, September 2006.
17. Sahika Genc and Stéphane Lafortune. Predictability of event occurrences in partially-observed discrete-event systems. *Automatica*, 45(2):301–311, February 2009.
18. Alejandro Grez, Filip Mazowiecki, Michał Pilipczuk, Gabriele Puppis, and Cristian Riveros. The monitoring problem for timed automata, 2020.
19. Léo Henry, Thierry Jéron, and Nicolas Markey. Control strategies for off-line testing of timed systems. In *SPIN*, pages 171–189, 06 2018.
20. Thomas A. Henzinger and Joseph Sifakis. The embedded systems design challenge. In Jayadev Misra, Tobias Nipkow, and Emil Sekerinski, editors, *Proceedings of the 14th International Symposium on Formal Methods (FM'06)*, volume 4085 of *Lecture Notes in Computer Science*, pages 1–15. Springer-Verlag, August 2006.
21. Anders Hessel, Kim Larsen, Marius Mikucionis, Brian Nielsen, Paul Pettersson, and Arne Skou. Testing Real-Time systems using UPPAAL. In Robert M. Hierons, Jonathan P.

Bowen, and Mark Harman, editors, *Formal Methods and Testing: An outcome of the FORTEST network*, volume 4949 of *Lecture Notes in Computer Science*, pages 77–117. Springer-Verlag, 2008.
22. Charles Antony Richard Hoare. An axiomatic basis for computer programming. *Communications of the ACM*, 12(10):576–580, October 1969.
23. Moez Krichen and Stavros Tripakis. Conformance testing for real-time systems. *Formal Methods in System Dessign*, 34(3):238–304, 2009.
24. Martin Leucker and Christian Schallart. A brief account of runtime verification. *Journal of Logic and Algebraic Programming*, 78(5):293–303, May 2009.
25. Meera Sampath, Raja Sengupta, Stéphane Lafortune, Kasim Sinnamohideen, and Demosthenis Teneketzis. Failure diagnosis using discrete-event models. *IEEE Transactions on Computers*, 35(1):105–124, January 1996.
26. Jan Tretmans. Conformance testing with labelled transition systems: Implementation relations and test generation. *Computer Networks and ISDN Systems*, 29(1):49–79, 1996.
27. Stavros Tripakis. Fault diagnosis for timed automata. In Werner Damm and Ernst Rüdiger Olderog, editors, *Formal Techniques in Real-Time and Fault-Tolerant Systems*, pages 205–221, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.
28. Stavros Tripakis. Folk theorems on the determinization and minimization of timed automata. *Information Processing Letters*, 99(6):222–226, September 2006.