Dynamical properties of timed automata revisited

Cătălin Dima

LACL, Université Paris 12, 61 av. du Général de Gaulle, 94010 Créteil, France

Abstract. We give a generalization of a solution by Puri to the problem of checking emptiness in timed automata with drifting clocks for the case of automata with non-closed guards. We show that non-closed guards pose certain specific problems which cannot be handled by Puri's algorithm, and propose a new algorithm, based on the idea of "boundary clock regions" of Alur, LaTorre and Pappas. We then give a symbolic algorithm for solving the reachability problem. Our algorithm is based on a symbolic construction of the "neighborhood" of a zone, and on a procedure that, given a set of zones $\mathcal{Z}$, builds the forward propagation of the strongly connected components which can be reached from $\mathcal{Z}$. This improves a symbolic algorithm of Daws and Kordy, due to the ability to handle sets of zones.

1 Introduction

Timed automata [AD94] are a widely accepted and powerful model of real time systems. They are finite automata endowed with dense-time variables, called *clocks*, that are used to measure time intervals separating actions. Dense time is utilized as an abstraction, in the sense that the model is not sensitive w.r.t. the "clock tick" in the implementation. Efficient algorithms and tools [LPY97,BDM$^+$98,HHWT97] have been designed and applied with succes for the verification of safety properties of systems modeled as timed automata.

Clocks in timed automata are synchronous: letting time pass t units increases *all* clocks with t. This assumption is sometimes too strong in distributed systems, in which some degree of non-synchronicity between the local clocks of each component may be present. The rectangular automata of [HKPV98] utilize dense variables that increase at a rate in some interval $[\alpha, \beta]$, and the subclass of initialized rectangular automata has a decidable reachability problem. Initialized rectangular automata may give a model of "inexact" timed automata with known clock drift Δ, if the rate binding interval is always $[1-\Delta, 1+\Delta]$. Hence, timed automata with a *known* clock drift have a decidable reachability problem.

In this paper, we are interested in solving the following *safe implementation problem:* given a timed automaton $\mathcal{A}$ and some zone Z, does there exist Δ for which no trajectory in which clocks drift with at most Δ units reaches Z? In [Pur98], A. Puri has shown how to compute the set of states that are reachable for any clock drift Δ. [Pur98] showed that, in the case of closed constraints, the region automaton does not provide the complete answer, as one needs to consider cycles in the (closed) region graph that have a nonempty intersection with regions that are already reachable. This result was further extended, in [DK06], where a symbolic algorithm for constructing a "stable zone" in the region graph is given. The stable zone is constructed for each cycle in the automaton graph that can be reached. Both the work in [Pur98] and in [DK06] treat of automata with closed clock constraints.

In this paper, we extend these results in two directions. First, we investigate the extension of the technique of Puri to automata with non-closed constraints. Secondly, we give a symbolic algorithm that constructs "sets of stable zones" as (forward propagations of) sets of zones which lie on some cycle in the region graph. Our technique can be applied to data structures for representing *sets of zones*, like in [LPWY99].

Our extension of Puri's technique to non-closed guards utilizes a variant of *boundary clock regions* of [ATP01], to model the fact that trajectories that pass through some region R can be arbitrarily close to some region R' neighboring R. Note that, as in [Pur98], we work with automata with bounded constraints; another assumption is that discrete transitions in a run are separated by non-zero delays.

Recently, [DDR05b,AT05] addressed a similar class of problems: given a timed automaton $\mathcal{A}$, does there exist some $\Delta > 0$ and a Δ-drift clock implementation of $\mathcal{A}$, in which "important" properties of $\mathcal{A}$ be preserved? [DDR05b,DDR05a] consider this problem in the context of verifying whether a controller C specified as a timed-automaton can be implemented by some drifting-clock automaton. Their approach is to model the system composed of the controller and the environment it must control as a parametric rectangular automaton in which Δ is a parameter. The drawback of this approach is that parametric model checking of timed automata with three clocks and only one parameter is known to be undecidable [AHV93,WT97]. Hence, tools like Uppaal cannot be directly applied to synthesize the value of Δ. The approach proposed in [DDR05a] is to guess an initial value for Δ and check it with Uppaal; if this guess satisfies the desired properties, then, according to the results of [DDR05b], any "faster" implementation (with $\Delta' < \Delta$) is also correct. This guess could be avoided by using the techniques from [DK06] and this paper.

The rest of the paper is divided as follows: in the next section we recall the definition and basic facts about timed automata and their drifting semantics. Section 3 contains the construction of the *boundary region automaton* and its correctness. Section 4 is devoted to the presentation of the symbolic algorithm and to comments on the improvements of our approach w.r.t. [DK06]. We end with a section with conclusions.

2 Timed automata

A **timed automaton** [AD94] is a tuple $\mathcal{A} = (Q, \mathcal{X}, \delta, Q_0, Q_f)$ where Q is a finite set of *locations*, $\mathcal{X}$ is a finite set of *clocks*, $Q_0, Q_f \subseteq Q$ are sets of *initial*, resp. *final* locations, and δ is a finite set of tuples called *transitions*, (q, C, X, q'), where $q, q' \in Q$, $X \subseteq \mathcal{X}$, and C is a finite conjunction of *simple constraints* utilizing clocks as variables – that is, constraints of the form $x \in I$, where I is an interval with nonnegative integer bounds. We will consider in this paper only *bounded* intervals, i.e. *excluding* intervals of the form $[2,\infty[$. For each $(q,C,X,r)\in\delta$, the component C is called the *guard* of the transition and X is its *reset component*. We consider the set of clocks is ordered as $\mathcal{X}=\{x_1,\ldots,x_n\}$.

In the standard semantics $\mathcal{A}$ can make time-passage transitions, in which all clocks advance with the same amount of time, and discrete transitions, in which location may change. The last are enabled when the "current clock valuation" satisfies the guard C of a transition (q,C,X,q'), and when they are executed, the clocks in the "reset component" X are set to zero. The notations used henceforth are the following: for a given point $v\in\mathbb{R}^n_{\geq 0}$ and $X\subseteq\mathcal{X}$, $v[X:=0]$ is the point obtained by reseting all clocks in X, defined by $(v[X:=0])_i=v_i$ for $x_i\notin X$ and $(v[X:=0])_i=0$ for $x_i\in X$. We will also denote $\mathbf{0}_n$ the origin point, i.e. $(\mathbf{0}_n)_i=0$ for all $1\leq i\leq n$.

In the drifting semantics [Pur98,DDR05b], when time advances by t, each clock advances with some $t'\in[t(1-\Delta),t(1+\Delta)]$, independently of the others, $\Delta>0$ denoting the maximal clock drift. The **Δ-drifting semantics** of $\mathcal{A}$ is the timed transition system $\mathcal{T}_\Delta(\mathcal{A})=(\mathcal{Q},\theta_\Delta,\mathcal{Q}_0,\mathcal{Q}_f)$ where $\mathcal{Q}=Q\times\mathbb{R}^n_{\geq 0}$, $\mathcal{Q}_0=Q_0\times\{\mathbf{0}_n\}$ (all clocks are set to *zero initially*), $\mathcal{Q}_f=Q_f\times\mathbb{R}^n_{\geq 0}$ and

$$\begin{aligned}\theta_\Delta=&\left\{(q,v)\xrightarrow{t}_\Delta(q,v')\mid t>0, v'_i-v_i\in\left[t(1-\Delta),t(1+\Delta)\right]\ \forall 1\leq i\leq n\right\}\\&\cup\left\{(q,v)\xrightarrow{\downarrow}_\Delta(q',v[X:=0])\mid\exists(q,C,X,q')\in\delta\text{ such that }v\models C\right\}\end{aligned}$$

Here $\models$ denotes the usual satisfiaction relation for clock constraints. Elements of $\mathcal{Q}$ are called **states**. When the automaton $\mathcal{A}$ is fixed, we use $\mathcal{T}_\Delta$ for $\mathcal{T}_\Delta(\mathcal{A})$.

A **$\mathcal{T}_\Delta$-trajectory** is a sequence of transitions $\tau=\big((q_{i-1},v_{i-1})\xrightarrow{\xi_i}_\Delta(q_i,v_i)\big)_{1\leq i\leq k}$ in θ_Δ, with $\xi_i\in\mathbb{R}_{>0}\cup\{\downarrow\}$. We denote this situation as $(q_0,v_0)\overset{\tau}{\rightsquigarrow}_{\mathcal{T}_\Delta}(q_k,v_k)$. Also, we denote $(q,v)\rightsquigarrow_{\mathcal{T}_\Delta}(q',v')$ when there exists a $\mathcal{T}_\Delta$-trajectory τ such that $(q,v)\overset{\tau}{\rightsquigarrow}_{\mathcal{T}_\Delta}(q',v')$. Trajectory τ is **accepting** if it starts in $\mathcal{Q}_0$ and ends in $\mathcal{Q}_f$. The set of $\mathcal{T}_\Delta$-trajectories is denoted Traj_Δ.

A **run** in $\mathcal{A}$ is a sequence $\rho=\big((q_{i-1},C_i,X_i,q_i)\big)_{1\leq i\leq k}$ of transitions from δ. A run $\rho=\big((q_{i-1},C_i,X_i,q_i)\big)_{1\leq i\leq k}$ is **associated with** a $\mathcal{T}_\Delta$-trajectory $\tau=\big((\overline{q}_{i-1},v_{i-1})\xrightarrow{\xi_i}(\overline{q}_i,v_i)\big)_{1\leq i\leq l}$ if $l=2k$ or $l=2k+1$ and for each $1\leq i\leq k$, $\overline{q}_{2i}=\overline{q}_{2i+1}=q_i$, $\xi_{2i-1}\in\mathbb{R}_{>0}$, $\xi_{2i}=\downarrow$, $v_{2i}=v_{2i-1}[X_i:=0]$, $v_{2i-1}\models C_i$, and also $\xi_{2k+1}\in\mathbb{R}_{>0}$, $\overline{q}_0=\overline{q}_1=q_0$.

For each $\Delta>0$ and state $(q,v)\in\mathcal{Q}$, the set of **$\mathcal{T}_\Delta$-reachable states** from (q,v) is:

$$\mathsf{Reach}_\Delta(q,v)=\left\{(q',v')\in\mathcal{Q}\mid(q,v)\rightsquigarrow_{\mathcal{T}_\Delta}(q',v')\right\}$$

The **reachable states in the limit** from (q,v) are:

$$\mathsf{Reach}_{\Delta\to 0}(q,v)=\bigcap\nolimits_{\Delta>0}\mathsf{Reach}_\Delta(q,v)$$

By extension, for any $S\subseteq\mathbb{R}^n_{\geq 0}$, we denote $\mathsf{Reach}_\Delta(q,S)=\bigcup_{v\in S}\mathsf{Reach}_\Delta(q,v)$ and $\mathsf{Reach}_{\Delta\to 0}(q,S)=\bigcup_{v\in S}\mathsf{Reach}_{\Delta\to 0}(q,v)$.

Throughout this paper we assume that there exists a clock x which is reset at each transition and which is checked, on each transition, to be greater than zero. Note that this assumption implies the fact that each cycle in the timed automaton contains a clock reset, as in [Pur98]. We also consider that $\mathcal{A}$ has no self loops. Note also that the semantics of $\mathcal{T}_\Delta$ (in which time steps have non-zero duration) also implies that time must strictly progress within each cycle, as required in [DDMR04]. It is well-known that any timed automaton can be transformed syntactically into an automaton satisfying these assumptions.

Regions and region reachability. A **zone** [Yov98] is an n-dimensional convex set of points which can be uniquely represented by a diagonal constraint of the form $C_Z = \bigwedge_{0\leq i,j\leq n}(x_i - x_j \in I_{ij})$, where $x_0 = 0$ and I_{ij} are intervals with integer bounds satisfying the following *triangle inequality*: $\forall 1 \leq i,j,k \leq n, I_{ik} \subseteq I_{ij} + I_{jk}$. The constraint C_Z is called the **normal form representation** of Z.

For $M \in \mathbb{N}$, an M-**region** (or simply a **region**, when M is understood from the context) is a zone R for which the intervals in the normal form representation C_R are either point intervals $I_{ij} = \{a\}$ with $-M \leq a \leq M$, or open unit intervals $I_{ij} =]a, a+1[$ with $-M \leq a \leq M-1$ $(a \in \mathbb{N})$.

Remark 1. Throughout this paper $M_\mathcal{A}$ will denote the largest constant occurring in a constraint in $\mathcal{A}$. We denote $\mathsf{Reg}_\mathcal{A}$ the set of $M_\mathcal{A}$-regions.

The **region automaton** is then $\mathcal{R}_\mathcal{A} = \big(Q \times \mathsf{Reg}_\mathcal{A}, \delta_\mathcal{R}, \mathcal{R}_0, \mathcal{R}_f\big)$ where $\mathcal{R}_0 = \big\{(q, \mathbf{0}_n) \mid q \in Q_0\big\}$, $\mathcal{R}_f = \big\{(q, R) \mid q \in Q_F\big\}$ and

$$\begin{aligned}\delta_\mathcal{R} = \big\{(q,R) \xrightarrow{\mathtt{t}} (q,R') \mid R \neq R', \exists v \in R, v' \in R', t \in \mathbb{R}_{>0} \text{ s.t. } (q,v) \xrightarrow{t} (q,v') \\ \text{and } \forall 0 < t' < t, \forall v'' \in \mathbb{R}^n_{\geq 0} \text{ if } (q,v) \xrightarrow{t'} (q,v''), \text{ then } v'' \in R \cup R'\big\} \\ \cup \big\{(q,R) \xrightarrow{\downarrow} (q',R') \mid \exists v \in R, v' \in R', \text{ s.t. } (q,v) \xrightarrow{\downarrow} (q',v')\big\}\end{aligned}$$

$\xrightarrow{\mathtt{t}}$ denotes here the *immediate time successor relation*, the time successor relation from [AD94] is denoted $\xrightarrow{\mathtt{t}}{}^*$. A *run* in $\mathcal{R}_\mathcal{A}$ is a sequence of transitions from $\delta_\mathcal{R}$. Tuples $(q,R) \in Q \times \mathsf{Reg}_\mathcal{A}$ will be called *state regions*.

It is well-known [AD94] that the *region automaton* is a faithful representation of the set of reachable states of $T_0(\mathcal{A})$: there exists a reachable final state $(q,v) \in \mathcal{Q}_f$ iff there exists a reachable state region (q,R) in $Q \times \mathsf{Reg}_\mathcal{A}$ with $v \in R$.

Figure 1 gives an example of a timed automaton and its associated region automaton. The dashed line gives the only transition between a state region of the form (q_0, R) to a state region of the form (q_1, R). Note that no final region is reachable from $(q_0, \mathbf{0}_2)$ in this automaton.

For each state region R and location $q \in Q$ we denote $\mathsf{RegReach}_\Delta(q,R)$ the set of state regions (q',R') that can be touched by a trajectory of $\mathcal{T}_\Delta$ that starts in (q,R), with $\Delta > 0$ fixed; we also denote $\mathsf{RegReach}_{\Delta\to 0}(q,R)$ the set of state regions (q',R') for which, *for each* $\Delta > 0$, there exists a trajectory in $\mathcal{T}_\Delta$ starting in (q,R) that touches (q',R'); these notations are also extended to zones Z:

$$\begin{aligned}\mathsf{RegReach}_\Delta(q,R) &= \big\{(q',R') \mid \exists (q',v) \in \mathsf{Reach}_\Delta(q,R), v \in R'\big\} \\ \mathsf{RegReach}_{\Delta\to 0}(q,R) &= \bigcap\nolimits_{\Delta>0} \mathsf{RegReach}_\Delta(q,R) \\ \mathsf{RegReach}_\Delta(q,Z) &= \bigcup \big\{\mathsf{RegReach}_\Delta(q,R) \mid R \in \mathsf{Reg}_\mathcal{A}, R \subseteq Z\big\} \\ \mathsf{RegReach}_{\Delta\to 0}(q,Z) &= \bigcup \big\{\mathsf{RegReach}_{\Delta\to 0}(q,R) \mid R \in \mathsf{Reg}_\mathcal{A}, R \subseteq Z\big\}\end{aligned}$$

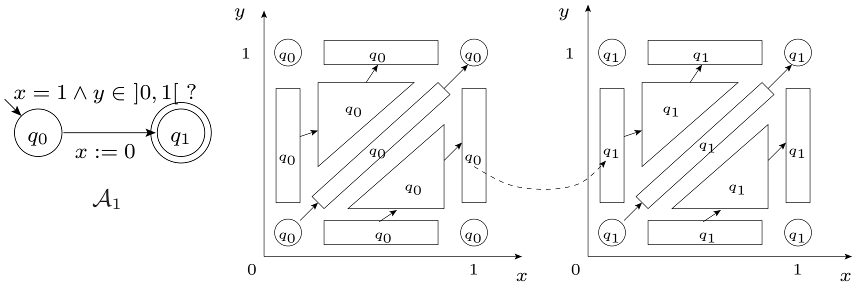

Fig. 1. The timed automaton $\mathcal{A}_1$ and its associated region automaton.

Example 1. Consider again the timed automaton in Figure 1 and the region R_1 defined by the constraint $C_{R_1} = 0 < x < y < 1$. Then, for any $\Delta > 0$, $\big\{(q_1, v_2) \mid (q_0, \mathbf{0}_2) \rightsquigarrow_\Delta (q_1, v_2)\big\} \cap \big(\{q_1\} \times R_1\big) \neq \emptyset$, and therefore $(q_1, R_1) \in \mathsf{RegReach}_\Delta(q_0, \mathbf{0}_2)$. This implies that $(q_1, R_1) \in \mathsf{RegReach}_{\Delta\to 0}(q_0, \mathbf{0}_2)$.

Note also that, in the region automaton for $\mathcal{A}_1$, the state region (q_1, R_1) is unreachable from $(q_0, \mathbf{0}_2)$, hence $\mathsf{RegReach}_{\Delta\to 0}(q_0, \mathbf{0}_2) \supsetneq \mathsf{RegReach}_0(q_0, \mathbf{0}_2)$.

Consider now the following *safe implementation problem*:

Problem 1. Given a zone Z and a location $q \in Q$, does there exist a clock drift Δ for which no trajectory in $\mathcal{T}_\Delta$ reaches a state (q, v) with $v \in Z$?

Note that the safe implementation problem is not equivalent with checking whether $\mathsf{Reach}_{\Delta\to 0}(q_0, \mathbf{0}_n) \cap (q, Z) \neq \emptyset$: in $\mathcal{A}_1$ from Figure 1, for any $\Delta > 0$, if $(q_0, \mathbf{0}_2) \xrightarrow{t}_\Delta (q_0, v_1) \xrightarrow{\downarrow}_\Delta (q_1, v_2)$ for some clock valuations $v_1, v_2 \in [0,1]^2$ then $v_1(y) = v_2(y) \in [1-\Delta, 1[$ and, therefore, for the region R_2 defined by $C_{R_2} = (x = 0 \wedge y = 1)$,

$$\begin{aligned}\mathsf{Reach}_{\Delta\to 0}(q_0, \mathbf{0}_2) \cap (q_1, R_2) &= \bigcap\nolimits_{\Delta>0}\big\{(q_1, v_2) \mid (q_0, \mathbf{0}_2) \rightsquigarrow_\Delta (q_1, v_2), v_2 \in R_2\big\} \\ &\subseteq \bigcap\nolimits_{\Delta>0}\{q_1\} \times \big(([0,1] \times [1-\Delta, 1[\,) \cap R_2\big) = \emptyset\end{aligned}$$

On the other hand, Example 1 above shows that $(q_1, R_1) \in \mathsf{RegReach}_{\Delta\to 0}(q_0, \mathbf{0}_2)$ and hence $\mathsf{RegReach}_{\Delta\to 0}(q_0, \mathbf{0}_2) \cap (q_1, R_2) \neq \emptyset$.

More interestingly, we may also observe that

$$\begin{aligned}&\mathsf{Reach}_{\Delta\to 0}(q_0, \mathbf{0}_2) \cap (q_1, [0,1]^2) \\ &\quad\subseteq \bigcap\nolimits_{\Delta>0}\{q_1\} \times \Big(\big(\{0\} \times [1-\Delta, 1[\,\big) \cup \big(]0, \frac{\Delta+\Delta^2}{1-\Delta}] \times [1-\Delta, 1]\big)\Big) = \emptyset\end{aligned}$$

which actually means that no state in the state region (q_1, R) can be Δ-reached for any Δ. Hence the pure study of $\mathsf{Reach}_{\Delta\to 0}$ is insufficient for solving the safe implementation problem.

The example with region R_2 also suggests that "closing the guards" in the given timed automata would not work. By closing the guards, we mean here the transformation of each automaton $\mathcal{A}$ into a timed automaton $\overline{\mathcal{A}}$ in which each transition copies a transition of $\mathcal{A}$, but with all constraints transformed into non-strict. To see that this technique does not work in general, note that, in $\overline{\mathcal{A}}_1$, $(q_1, R_2) \in \mathsf{Reach}_0(q_0, \mathbf{0}_n) \setminus \mathsf{RegReach}_{\Delta\to 0}(q_0, \mathbf{0}_n)$.

Before ending this section, we give a useful technical property relating convexity with runs in the region automaton. This property utilizes the following notion of *association* between runs in the timed automaton and runs in the region automaton: a run $\rho = ((q_{i-1}, R_{i-1}) \xrightarrow{\xi_i} (q_i, R_i))_{1\le i\le k}$ in $\mathcal{R}_\mathcal{A}$ is *associated* with a run $\rho' = \big((r_{j-1}, C_j, X_j, r_j)\big)_{1\le i\le m}$ in $\mathcal{A}$ if there are m indices $j_1, \dots, j_m \le k$ such that $\xi_{j_l} =\downarrow$, $R_{j_l} = R_{j_{l-1}}[X := 0]$ and $R_{j_{l-1}} \subseteq Z_{C_l}$ $(1 \le l \le m)$, and also $\xi_i = \mathtt{t}$ for all $i \ne j_1, \dots, j_m$. In other words, ρ and ρ' are associated iff all trajectories subsumed by ρ are associated with ρ'.

For each bounded region $R \in \mathsf{Reg}_\mathcal{A}$, we denote by $V(R)$ the set of vertices, or *cornerpoints* that bound R. For example, for the 2-dimensional region $0 < x < y < 1$, $V(R) = \{(0,0), (0,1), (1,1)\}$. Note that $V(R)$ can be formally defined using the "fractional part" [AD94] representation of regions.

Proposition 1. *Suppose ρ_1 and ρ_2 are two runs in $\mathcal{R}_\mathcal{A}$, with ρ_i starting in (q, R_i') and ending in (q', R_i'') $(i = 1, 2)$. Suppose that both runs are associated with the same run ρ in $\mathcal{A}$ and that there exist R, R' such that $V(R) = V(R_1) \cup V(R_2)$, $V(R') = V(R_1') \cup V(R_2')$. Then there is a run ρ' in $\mathcal{R}_\mathcal{A}$ that starts in (q, R_1), ends in (q', R_2) and is associated with ρ.*

The proof of this property is based on zone convexity.

3 The extended boundary region automaton

First, let us recall here briefly Puri's technique for constructing the set of reachable regions in a closed timed automaton: the $\Delta \to 0$-reachable regions are obtained by applying, alternatively, the following two procedures until a fixpoint is reached:

1. Forward closure of a given set of regions.
2. Add cycles in the "closed region automaton", that have a nonempty intersection with an already reachable region,

In the previous section, we have seen that this technique cannot be applied as is to the closure $\overline{\mathcal{A}}$ of the given automaton $\mathcal{A}$, due to inherent peculiarities of working with non-closed guards.

In this section, we refine Puri's technique of searching for cycles in the region graph, by carefully defining when to consider that a (possibly open) region "touches" a reachable region. The right notion of "touching" is given in the following definition:

Definition 1. *A region R is **$\mathtt{t}$-aligned** if its normal form representation $C_R = \bigwedge_{0\le i,j\le n} x_i - x_j \in I_{ij}$ has the property that I_{i0} is not a point interval for all i. Equivalently, for any $q \in Q$, there exist $v, v' \in R$ with $(q, v) \xrightarrow{t}_0 (q, v')$ for some $t > 0$.*

*Two regions R, R' are **neighbors** if $V(R) \cap V(R') \ne \emptyset$. R, R' are **$\mathtt{t}$-neighbors** if both are $\mathtt{t}$-aligned and either $V(R') \subseteq V(R)$ or $V(R) \subseteq V(R')$.*

Example 2. For any timed automaton, the region R_1 with $C_{R_1} : 0<x<y<1$ is a $\mathtt{t}$-neighbor for R_2, with $C_{R_2} : 0<x=y<1$, while $\mathbf{0}_2$ is not a $\mathtt{t}$-neighbor of R_2.

The idea behind the computation of $\mathsf{RegReach}_{\Delta\to 0}$ is to utilize, instead of regions, pairs of regions (R, R') with $V(R) \supseteq V(R')$. Such pairs are similar to the *boundary regions* of [ATP01]: they model sets of trajectories that pass through R and are "arbitrarily close" to R'. Formally we construct the *boundary regions automaton* $\mathcal{E}(\mathcal{A}) = \{\mathcal{Q}_{\mathcal{E}(\mathcal{A})}, \theta_{\mathcal{E}(\mathcal{A})}, \mathcal{Q}_0^b, \mathcal{Q}_f^b\}$ where $\theta_{\mathcal{E}(\mathcal{A})} \subseteq \mathcal{Q}_{\mathcal{E}(\mathcal{A})} \times \mathcal{Q}_{\mathcal{E}(\mathcal{A})}$ and

$$\mathcal{Q}_{\mathcal{E}(\mathcal{A})} = \{(q, R, R') \mid q \in Q, R, R' \in \mathsf{Reg}, V(R) \supseteq V(R')\}$$

$$\mathcal{Q}_0^b = \{(q, \mathbf{0}_n, \mathbf{0}_n) \in \mathcal{Q}_{\mathcal{E}(\mathcal{A})} \mid q \in Q_0\} \text{ and } \mathcal{Q}_f^b = \{(q, R, R') \in \mathcal{Q}_{\mathcal{E}(\mathcal{A})} \mid q \in Q_f\}$$

$$\begin{aligned}\theta_{\mathcal{E}(\mathcal{A})} = & \{(q, R_1, R) \xrightarrow{n}_{\mathcal{E}(\mathcal{A})} (q, R_2, R) \mid R_1, R_2 \text{ are } \mathtt{t}\text{-neighbors}\} \\ & \cup \{(q, R_1, R) \xrightarrow{\mathtt{t}_1}_{\mathcal{E}(\mathcal{A})} (q, R_2, R) \mid R_1 \xrightarrow{\mathtt{t}} R_2\} \\ & \cup \{(q, R, R_1) \xrightarrow{r}_{\mathcal{E}(\mathcal{A})} (q, R, R_2) \mid V(R_1) \supseteq V(R_2)\} \\ & \cup \{(q, R, R_1) \xrightarrow{\mathtt{t}_2}_{\mathcal{E}(\mathcal{A})} (q, R, R_2) \mid R_1 \xrightarrow{\mathtt{t}} R_2\} \\ & \cup \{(q_1, R_1, R_1') \xrightarrow{\downarrow}_{\mathcal{E}(\mathcal{A})} (q_2, R_2, R_2') \mid (q_1, R_1) \xrightarrow{\downarrow} (q_2, R_2) \in \delta_{\mathcal{R}} \text{ and} \\ & \qquad \text{if } R_2 = R_1[X := 0] \text{ for some } X \subseteq \mathcal{X}, \text{ then } R_2' = R_1'[X := 0]\} \\ & \cup \{(q, R_1, R_1') \xrightarrow{\circlearrowleft}_{\mathcal{E}(\mathcal{A})} (q, R_2, R_2') \mid R_1', R_2' \text{ are } \mathtt{t}\text{-neighbors}, ((q, R_2'), (q, R_2')) \in \delta_{\mathcal{R}}^+\}\end{aligned}$$

($\delta_{\mathcal{R}}^+$ is the transitive closure of the transition relation in the region automaton.)

Elements of $\mathcal{Q}_{\mathcal{E}(\mathcal{A})}$ will be called *boundary regions*. Each type of transition in $\mathcal{E}(\mathcal{A})$ is labeled differently, due to its particular significance: transitions $\xrightarrow{n}$ are between $\mathtt{t}$-**n**eighboring boundary regions, $\xrightarrow{\mathtt{t}_1}$ and $\xrightarrow{\mathtt{t}_2}$ are the two types of $\mathtt{time}$-passage transitions, while $\xrightarrow{r}$ represent **r**eductions to a smaller boundary region. The reflexive-transitive closure of $\theta_{\mathcal{E}(\mathcal{A})}$ will be denoted $\to_{\mathcal{E}(\mathcal{A})}$, while subsets of it involving only certain types of transitions are identified by their respective symbols. For example, $\xrightarrow{\mathtt{t}_1,\mathtt{t}_2,n}_{\mathcal{E}(\mathcal{A})} = \left(\xrightarrow{\mathtt{t}_1}_{\mathcal{E}(\mathcal{A})} \cup \xrightarrow{\mathtt{t}_2}_{\mathcal{E}(\mathcal{A})} \cup \xrightarrow{n}_{\mathcal{E}(\mathcal{A})} \right)^*$.

We will say that the transition $(q_1, R_1, R_1') \xrightarrow{\downarrow}_{\mathcal{E}(\mathcal{A})} (q_2, R_2, R_2')$ is *associated with* a transition $\tau = (q_1, C, X, q_2) \in \delta$ if $R_1 \subseteq Z_C$ and $R_2 = R_1[X := 0]$.

An example is provided in Figure 2 for $\mathcal{E}(\mathcal{A}_1)$, where $\mathcal{A}_1$ is the automaton from Figure 1. (Some of the transitions are labeled only with one of the types they may carry.) Note that, starting from $(q_0, \mathbf{0}_2, \mathbf{0}_2)$, the only reachable boundary regions in $\mathcal{E}(\mathcal{A}_1)$ in which location q_1 occurs are of the type (q_1, R, R') in which $C_{R'} : (x = 0 \wedge y = 1)$ and $R \neq R'$. As we will see, this is consistent with the fact that $\mathsf{RegReach}_{\Delta\to 0}(q_0, \mathbf{0}_2) \not\ni (q_1, R'')$ where $C_{R''} : (x = 0 \wedge y = 1)$.

The following property says that the third components in boundary regions always follow the transitions of the closure $\overline{\mathcal{A}}$ of the given timed automaton $\mathcal{A}$:

Lemma 1. *If* $(q_1, R_1, R_1') \xrightarrow{n,\mathtt{t}_1,\mathtt{t}_2,r,\downarrow}_{\mathcal{E}(\mathcal{A})} (q_2, R_2, R_2')$ *then* $((q_1, R_1'), (q_2, R_2')) \in \delta_{\overline{\mathcal{A}}}^*$.

The proof is by straightforward induction on the length of the $\mathcal{E}(\mathcal{A})$-run.

Denote $\mathsf{Reach}_{\mathcal{E}(\mathcal{A})}(q_0, \mathbf{0}_n, \mathbf{0}_n)$ the set of boundary regions that can be reached from $(q_0, \mathbf{0}_n, \mathbf{0}_n)$ in $\mathcal{E}(\mathcal{A})$. The first main result of this paper is the following:

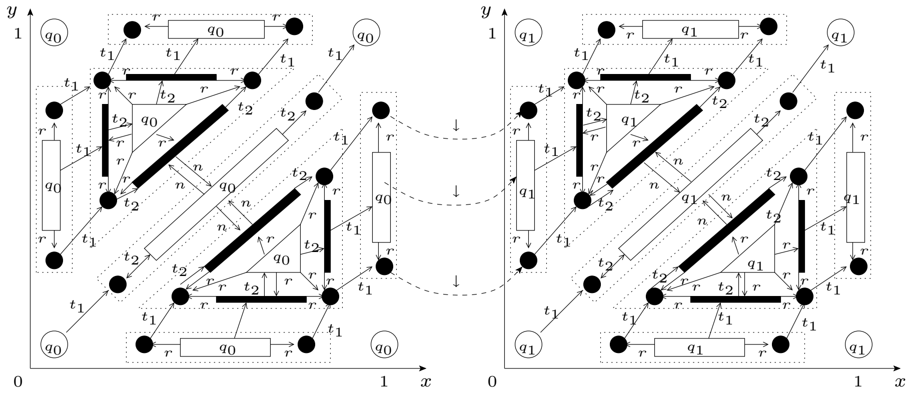

Fig. 2. $\mathcal{E}(\mathcal{A}_1)$ for the automaton in Figure 1.

Theorem 1. *Let $\mathcal{A}$ be a bounded timed automaton with no self loops and in which there exists a clock x such that all transitions (q, C, X, q') have $x \in X$ and $C \wedge (x = 0)$ not satisfiable. Then:*

$$\mathsf{RegReach}_{\Delta\to 0}(q_0, \mathbf{0}_n) = \big\{(q, R) \mid \exists R' \in \mathsf{Reg}_{\mathcal{A}}, (q, R, R') \in \mathsf{Reach}_{\mathcal{E}(\mathcal{A})}(q_0, \mathbf{0}_n, \mathbf{0}_n)\big\}$$

The inverse inclusion is a corollary of the following technical property:

Proposition 2. *Suppose $(q_1, R_1, R_1') \to_{\mathcal{E}(\mathcal{A})} (q_2, R_2, R_2')$ and denote $d(v, v') = \max |v_i - v_i'|$ (i.e. the max-distance), and also $d(v, R) = \min\{d(v, v') \mid v' \in R\}$.*

Then for all $\Delta > 0$ and $0 < \eta < \Delta$ there exists $\zeta \leq \eta$ such that for all $v_2 \in R_2$ for which $d(v_2, R_2') < \zeta$ there exists $v_1 \in R_1$ for which $d(v_1, R_1') < \eta$ and with $(q_1, v_1) \xrightarrow{t}_{\Delta} (q_2, v_2)$ for some $t \in \mathbb{R}_{>0}$.

The proof is by induction on the number of transitions in $\mathcal{E}(\mathcal{A})$. There are 6 base cases, according to the types of $\mathcal{E}(\mathcal{A})$-transition. The case $(q, R_1, R_1') \xrightarrow{\circlearrowleft}_{\mathcal{E}(\mathcal{A})} (q, R_2, R_2')$ relies on the following easy generalization of Theorem 7.3 of [Pur98]:

Lemma 2. *Given $R \in \mathsf{Reg}_{\mathcal{A}}$ and $q \in Q$ with $((q, R), (q, R)) \in \delta_{\mathcal{R}}^+$, then for any $v, v' \in R$ and for any $\Delta > 0$, $(q, v') \in \mathsf{Reach}_{\Delta}(q, v)$.*

The following straightforward corollary of Lemma 2 will be essential in the construction of our symbolic algorithm:

Corollary 1. *Suppose that the state regions $(q_1, R_1) \neq (q_2, R_2)$ belong to the same strongly connected component in the region automaton, and $((q_2, R_2), (q_3, R_3)) \in \delta_{\mathcal{R}}^*$. Then for any $v_1 \in R_1$, $v_3 \in R_3$ and for any $\Delta > 0$, $(q_3, v_3) \in \mathsf{Reach}_{\Delta}(q_3, v_3)$.*

Note that the validity of this corollary relies on the fact that we only consider bounded regions.

The proof of the direct inclusion in Theorem 1 relies on a "continuous" presentation of trajectories. In the sequel, for a mapping $f : A \to B \times C$, $f\big|_2 : A \to C$ denotes the second projection. Also, for a real function $f : I \to A$ with $I \subseteq \mathbb{R}_{\geq 0}$ and $J \subseteq I$, $f\big|_J$ is the usual restriction of f to J.

Definition 2. *A **continuous Δ-trajectory** ($\Delta \in \mathbb{R}_{\geq 0}$) is a mapping $\phi : [0, \alpha[\to Q \times \mathbb{R}^n_{\geq 0}$ satisfying the following properties:*

1. *For each $q \in Q$, $\phi^{-1}(q \times \mathbb{R}^n_{\geq 0})$ is a finite union of left-closed, right-open intervals $\left(I^i_{\phi,q}\right)_{1 \leq i \leq n_{\phi,q}}$, with $I^i_{\phi,q} = [\alpha^i_q, \alpha^{i+1}_q[$.*
2. *For any two distinct states $q, r \in Q, q \neq r$, and any two adjacent intervals $I^i_{\phi,q}, I^j_{\phi,r}$ (i.e., $\alpha^{i+1}_q = \alpha^j_r$), there exists a transition $(q, C, X, r) \in \delta$ which creates the "jump" from $I^i_{\phi,q}$ to $I^j_{\phi,r}$ in the following sense: if we denote $v = \lim_{x \nearrow \alpha^{i+1}_q} \phi\big|_2(x)$ and $v' = \phi\big|_2(\alpha^j_r)$, then $v \models C$ and $v' = v[X := 0]$.*
3. *For each $q \in Q$ for which $n_{\phi,q} > 0$, for each $1 \leq i \leq n_{\phi,q}$ and each $t, t' \in I^i_{\phi,q}$ with $t < t'$, there exists $u \in \mathbf{B}^n$ such that $\phi\big|_2(t') = \phi\big|_2(t) + (t' - t)(\mathbf{1} + \Delta u)$.*

*The continuous Δ-trajectory ϕ is **canonical** if the following property holds:*

4. *For each $t, t' \in [0, \alpha[$, if there exists $R \in \mathsf{Reg}_{\mathcal{A}}$ and $q \in Q$ such that $\phi(t), \phi(t') \in \{q\} \times R$ and for all t'' with $t \leq t'' \leq t'$, $\phi(t'') \in \{q\} \times \mathbb{R}^n_{\geq 0}$, then for all $t \leq t'' \leq t'$, $\phi(t'') \in \{q\} \times R$.*

The second property holds due to the assumption which forbids taking two discrete transitions without letting time pass. The fourth also is consistent since we only consider automata without self loops.

Canonical continuous trajectories avoid "volutes" between $\mathtt{t}$-neighbors. The following property shows that each Δ-trajectory, which gives only "essential points" through the behavior of a system, can be associated with a canonical continuous trajectory, which in fact completes the Δ-trajectory with all the intermediary points:

Proposition 3. *For each Δ-trajectory $\tau = \left((q_{i-1}, v_{i-1}) \xrightarrow{\xi_i} (q_i, v_i)\right)_{1 \leq i \leq k}$ ($\Delta \geq 0$) there exists a canonical continuous Δ-trajectory $\phi : [0, \alpha[\to Q \times \mathbb{R}^n_{\geq 0}$ for which $\phi(0) = (q_0, v_0)$, $\lim_{t \nearrow \alpha} \phi(t) = (q_k, v_k)$, and for each $1 \leq i < k$ there exists α_i with $\alpha_{i-1} < \alpha_i$ such that $\phi(\alpha_i) = (q_i, v_i)$.*

The proof follows by easy induction on the the length k of τ.

The following property states that, if we "simulate" the behavior of a Δ-trajectory ϕ with a "pseudo"-0-trajectory ϕ', then the final points in the two trajectories cannot be "too far" one from the other. In some sense, the simulating pseudo-0-trajectory models what would happen in $\overline{\mathcal{A}}$, if we were to "follow" the same run that is associated with ϕ, take the transitions at the same time points but without any clock drift and without checking any guard on the transitions (i.e. just resetting clocks).

Proposition 4. *Given a canonical continuous Δ-trajectory $\phi : [0, \alpha[\to Q \times \mathbb{R}^n_{\geq 0}$, consider a mapping $\phi' : [0, \alpha[\to Q \times \mathbb{R}^n_{\geq 0}$ with $\phi'(0) = \phi(0)$ and satisfying the following properties:*

1. *For all $q \in Q, 1 \leq i \leq n_{\phi,q}, t, t' \in I^i_{\phi,q}$ with $t < t'$, $\phi'(t') = \phi'(t) + t' - t$.*

2. *For any two states $q, r \in Q, q \neq r$ and any two adjacent intervals $I^i_{\phi,q}, I^j_{\phi,r}$ (with $\alpha^{i+1}_q = \alpha^j_r$), if (q, C, X, r) is the transition for which condition 2 in Definition 2 is met for ϕ, and we denote $v = \lim_{x \nearrow \alpha^{i+1}_q} \phi'\big|_2(x)$ and $v' = \phi'\big|_2(\alpha^j_r)$, then $v' = v[X := 0]$.*

Then for each $t \in [0, \alpha[$, $d(\phi\big|_2(t), \phi'\big|_2(t)) \leq t\Delta$.

Recall that d denotes the max-distance. The proof can be again given by induction on the number of regions through which ϕ passes.

Note that, by the construction in Proposition 4, there exists a unique mapping ϕ' associated to ϕ – we will call it the **0-approximation of** ϕ. ϕ' is not really a 0-trajectory since in condition 2 above we may have $v \not\models C$.

For the following lemma, we denote $\xrightarrow{\mathtt{t},0}$ the union of the identity relation with the immediate successor relation in $\mathcal{R}_\mathcal{A}$. The result here is needed when showing that regions that are "arbitrarily close" are forward-propagated through the same types of region transitions, then we obtain also regions that are "arbitrarily close":

Lemma 3. *Consider two tuples of regions $R_1, \ldots, R_n, R'_1, \ldots R'_n$ and two extra regions R, R' such that $V(R) = \bigcap_{1\leq i\leq n} V(R_i)$ and $V(R') = \bigcap_{1\leq i\leq n} V(R'_i)$.*

1. *Suppose that $(q, R_i) \xrightarrow{\mathtt{t},0} (q, R'_i)$ for all $1 \leq i \leq n$ and some $q \in Q$. Then $(q, R) \xrightarrow{\mathtt{t},0} (q, R')$.*
2. *If $(q, R_i) \xrightarrow{\downarrow} (q', R'_i)$ for all $1 \leq i \leq n$ and some $q, q' \in Q$, then $(q, R) \xrightarrow{\downarrow} (q', R')$.*

The first result follows by observing how linear combinations of vertices of a region evolve during time steps, whereas the second is straightforward.

The following technical property is needed when proving a somewhat reverse of the previous lemma: if two regions can be reached from one another via some (sufficiently small) time-passage transition with some drift $\Delta > 0$, and they are neighbors of some regions that can be reached from one another in the 0-drift region automaton, then, altogether, the four regions form a transition in $\mathcal{E}(\mathcal{A})$:

Proposition 5. *Given $\Delta > 0$, two regions $R_1 \neq R_2$, and a (canonical) continuous Δ-trajectory $\phi : [0, \alpha[\rightarrow Q \times \mathbb{R}^n_{\geq 0}$, suppose that $\phi(t) \in \{q\} \times (R_1 \cup R_2)$ for some $q \in Q$ and all $t \in [0, \alpha[$, and also that $\phi(0) \in (q, R_1)$, $\lim_{t \nearrow \alpha} \phi(t) \in (q, R_2)$. Then there exist R'_1, R'_2 such that $V(R'_1) \subseteq V(R_1)$ and $V(R'_2) \subseteq V(R_2)$ such that $(q, R'_1) \xrightarrow{\mathtt{t}} (q, R'_2)$. Moreover, $(q, R_1, R'_1) \rightarrow_{\mathcal{E}(\mathcal{A})} (q, R_2, R'_2)$.*

Together, Lemma 3 and Proposition 5 show how the neighborship relation between regions is related with the transition relation in the region automaton.

The final step in the proof of the direct inclusion in Theorem 1 is the following proposition. Here, we denote $T = card(\theta_{\mathcal{E}(\mathcal{A})})$ and $K = 2^{3n+1}$. Also $\mathbf{A}^{K+2}_{T+2} = \frac{(T+2)!}{(T-K)!}$ is the number of ordered tuples of $2^{3n+1} + 2$ elements from a set of $card(\theta_{\mathcal{E}(\mathcal{A})}) + 2$ elements. Note that $K < T$ for any automaton $\mathcal{A}$.

Proposition 6. *Take* $\phi : [0, \alpha[\to Q \times \mathbb{R}^n_{\geq 0}$ *a canonical continuous* Δ*-trajectory, with* $\Delta < \frac{1}{2(\mathbf{A}_{T+2}^{K+2})^2+2} \cdot \sqrt{\frac{2}{n}}$. *Denote* $\big(q_i, R_i\big)_{0 \leq i \leq m_\phi}$ *the sequence of state regions to which the points in* ϕ *belong, that is, if* $\phi(t) \in (q_i, R_i), \phi(t') \in (q_{i+1}, R_{i+1})$ *then* $\forall t'' \in]t, t'[, \phi(t'') \in (q_i, R_i) \cup (q_{i+1}, R_{i+1})$. *Then there exist regions* $(\tilde{R}_i)_{0 \leq i \leq m_\phi}$ *such that* $(q_{i-1}, R_{i-1}, \tilde{R}_{i-1}) \to_{\mathcal{E}(\mathcal{A})} (q_i, R_i, \tilde{R}_i)$ *for all* $1 \leq i \leq m_\phi$.

The proof works in two steps: first, for $\alpha < 2(\mathbf{A}_{T+2}^{K+2})^2 + 2$, we may show that there exist regions $(\tilde{R}_i)_{0 \leq i \leq m_\phi}$ such that $(q_{i-1}, R_{i-1}, \tilde{R}_{i-1}) \xrightarrow{\mathtt{t}_1, \mathtt{t}_2, n, r, \downarrow}_{\mathcal{E}(\mathcal{A})} (q_i, R_i, \tilde{R}_i)$ for all $1 \leq i \leq m_\phi$. Then, for the case of $\alpha \geq 2(\mathbf{A}_{T+2}^{K+2})^2 + 2$, we may show that, after most $2(\mathbf{A}_{T+2}^{K+2})^2 + 2$ transitions, the trajectory must pass through the same state region (q, R), which implies that, after the correspoding sequence of $\xrightarrow{\mathtt{t}_1, \mathtt{t}_2, n, r, \downarrow}_{\mathcal{E}(\mathcal{A})}$-transitions (as constructed in the first part), we may insert a $\circlearrowright$ transition. The whole argument is based on Lemmas 1, 3 (and some extra technical lemmas) and Propositions 1 and 5. The details of this proof can be found in the report [Dim06].

4 Symbolic computation of $\mathsf{RegReach}_{\Delta \to 0}(Q_0, \mathbf{0}_n)$

The idea behind our symbolic algorithm is to alternate forward reachability, $\mathtt{t}$-neighbor construction and cycle construction, until a fixpoint is reached. The cycle construction takes advantage of the special form of boundary regions in $\mathcal{E}(\mathcal{A})$ that give the possibility to obtain, symbolically, all regions that are neighbors of reachable regions. Our algorithm uses triples of the form (q, Z, Z') where $q \in Q$ and Z, Z' are zones with $Z' = \overline{Z'} \subseteq Z$. The algorithm generates triples (q, Z, Z') for which, for any regions R, R', if $R \subseteq Z$, $R' \subseteq Z'$ and $V(R) \supseteq V(R')$, then $(q_0, \mathbf{0}_n, \mathbf{0}_n) \to_{\mathcal{E}(\mathcal{A})} (q, R, R')$. But let us introduce first some notations.

The term $\mathcal{A}$-*zones* denotes zones $Z \subseteq [0, M_\mathcal{A}]^n$ (recall $M_\mathcal{A}$ is the maximal constant used in $\mathcal{A}$). The set of $\mathcal{A}$-zones is denoted $\mathcal{Z}_\mathcal{A}$. We also say that an $\mathcal{A}$-zone Z is *time passage closed* if for all $v \in Z$ and $q \in Q$, if $(q, v) \xrightarrow{t} (q, v')$ then $v' \in Z$. For any $\mathcal{A}$-zone Z we denote

$$\mathtt{t}\text{–}\mathsf{Nghbr}(Z) = Z \cup \bigcup \{R' \mid R \subseteq Z, R \in \mathsf{Reg}_\mathcal{A}, R, R' \text{ are } \mathtt{t}\text{-neighbors}\}$$

We also denote $\mathtt{t}\text{–}\mathsf{Nghbr}(q, Z) = \{q\} \times \mathtt{t}\text{–}\mathsf{Nghbr}(Z)$ for any $q \in Q$.

The following proposition gives a set of properties that characterize the $\mathtt{t}\text{–}\mathsf{Nghbr}$ operator and relate it with the boudary regions in $\mathcal{E}(\mathcal{A})$:

Proposition 7. *1. Suppose* Z *is an* $\mathcal{A}$*-zone, with* $C_Z = \bigwedge_{0 \leq j < i \leq n} x_i - x_j \in I_{ij}$. *Then* $\mathtt{t}\text{–}\mathsf{Nghbr}(Z)$ *is defined by the constraint in normal form*

$$C_{\mathtt{t}\text{–}\mathsf{Nghbr}(Z)} : \bigwedge_{1 \leq i \leq n} (x_i \in I_{i0}) \wedge \bigwedge_{1 \leq j < i \leq n} \Big(x_i - x_j \in \big(I_{ij} + \,]{-1}, 1[\big) \cap [-M_\mathcal{A}, M_\mathcal{A}] \Big)$$

2. Moreover, if Z *is time passage closed, then for all* $R_1, R_1' \subseteq Z$ *with* $V(R_1) \supseteq V(R_1')$, $(q, R_1, R_1') \xrightarrow{\mathtt{t}_1, \mathtt{t}_2, r, n}_{\mathcal{E}(\mathcal{A})} (q, R_2, R_2')$ *if and only if there exists* $(q, Z') \subseteq \mathtt{t}\text{–}\mathsf{Nghbr}(Z)$ *for which* $R_2, R_2' \subseteq Z'$ *and* $V(R_2) \supseteq V(R_2')$.

3. *Finally, suppose Z_1, Z_2 are two $\mathcal{A}$-zones which are time passage closed. Then $Z_2 = \mathtt{t}\text{–}\mathsf{Nghbr}(Z_1)$ if and only if for any region $R_1 \subseteq Z_1$ that is $\mathtt{t}$-aligned, and for any region R_2 which is a $\mathtt{t}$-neighbor for R_1, we have that $R_2 \subseteq Z_2$.*

Given a transition $\tau = (q, X, C, q')$ and a zone Z, the *forward propagation of (q, Z) along τ* is:

$$\mathbf{Fwd}(q, Z, \tau) = \big\{(q', Z') \mid \exists v \in Z, v' \in Z', v'' \in \mathbb{R}^n_{\geq 0}, t, t' \in \mathbb{R}_{\geq 0} \text{ s.t. } (q, v) \xrightarrow{t}_0 (q, v'') \xrightarrow{\downarrow}_0 (q', v''[X := 0]) \xrightarrow{t'}_0 (q', v') \text{ and } v'' \models C\big\}$$

The forward propagation of (q, Z) is then

$$\mathbf{Fwd}(q, Z) = \mu X.\big((q, Z) \cup \bigcup_{\tau \in \delta} \mathbf{Fwd}(X, \tau)\big)$$

It is well known [Yov98] that, for any zone Z, state q and transition τ, $\mathbf{Fwd}(q, Z, \tau)$ and $\mathbf{Fwd}(q, Z)$ are computable symbolically if there is no diagonal constraint in the given automaton $\mathcal{A}$ (see [BLR05]) – which is the case here.

We may then see that forward reachability and $\mathtt{t}$-neighborhoodness are related by the following property:

$$\mathtt{t}\text{–}\mathsf{Nghbr}(q, Z) = \bigcup \big\{(q, R) \mid \exists (q, R_1) \xrightarrow{\mathtt{t}}{}^{*} (q, R_2) \text{ in } \mathcal{R}_{\mathcal{A}}, \exists R_3 \in \mathsf{Reg}_{\mathcal{A}} \text{ s.t. } R_3 \subseteq \mathtt{t}\text{–}\mathsf{Nghbr}(R_1) \cap Z, R \subseteq \mathtt{t}\text{–}\mathsf{Nghbr}(R_2)\big\}$$

For each transition $\tau = (q_1, C, X, q_2) \in \delta$, we denote $\overline{\tau} = (q_1, \overline{C}, X, q_2)$ the *closure* of τ, with $\overline{C}$ being the closure of C (in the sense that all inequalities in C are transformed into nonstrict). Define then the *forward propagation* of a triple (q, Z, Z') along the transition τ as:

$$\mathbf{FwBnR}(q_1, Z_1, Z_1', \tau) := \big\{(q_2, Z_2, Z_2') \mid (q_2, Z_2') \in \mathbf{Fwd}(q_1, Z_1', \overline{\tau}), Z_2 \in \mathtt{t}\text{–}\mathsf{Nghbr}(Z_2')\big\}$$

The following proposition shows that the forward propagations of boundary regions can be computed symbolically, using the **FwBnR** operator:

Proposition 8. *Given $q_1 \in Q$ and Z_1, Z_1' two $\mathcal{A}$-zones which are time passage closed and with $Z_1 = \mathtt{t}\text{–}\mathsf{Nghbr}(Z_1')$. Then $(q_2, R_2, R_2') \in \mathbf{FwBnR}(q_1, Z_1, Z_1', \tau)$ if and only if there exist $(q_1, R_1, R_1') \in \mathcal{Q}_{\mathcal{E}(\mathcal{A})}$ with $R_1 \subseteq Z_1$ $R_1' \subseteq Z_1'$, and $(q_3, R_3, R_3'), (q_4, R_4, R_4') \in \mathcal{Q}_{\mathcal{E}(\mathcal{A})}$ such that*

$$(q_1, R_1, R_1') \xrightarrow{\mathtt{t}_1, \mathtt{t}_2, n, r}_{\mathcal{E}(\mathcal{A})} (q_3, R_3, R_3') \xrightarrow{\downarrow}_{\mathcal{E}(\mathcal{A})} (q_4, R_4, R_4') \xrightarrow{\mathtt{t}_1, \mathtt{t}_2, n, r} (q_2, R_2, R_2')$$

where the middle transition is associated with τ.

In the sequel, we will consider each set of tuplpes $\mathcal{Z} \subseteq Q \times \mathcal{Z}_{\mathcal{A}}$ as the set of state regions belonging to $\mathcal{Z}$. Hence, we abuse notation and denote $\mathcal{Z}_1 = \mathcal{Z}_2$ if $\{(q, R) \mid \exists (q, Z) \in \mathcal{Z}_1, R \subseteq Z\} = \{(q, R) \mid \exists (q, Z) \in \mathcal{Z}_2, R \subseteq Z\}$.

For each set of triples $\Upsilon \subseteq Q \times \mathcal{Z}_{\mathcal{A}} \times \mathcal{Z}_{\mathcal{A}}$, we define $\mathbf{FwBnR}(\Upsilon)$ as the set of triples (q, Z, Z') which can be reached from Υ by repeatedly applying forward propagation steps,

$$\mathbf{FwBnR}(\Upsilon) = \mu X.\big(\Upsilon \cup \bigcup_{\tau \in \delta} \bigcup_{(q, Z, Z') \in X} \mathbf{FwBnR}(q, Z, Z', \tau)\big)$$

This set can be computed as a fixpoint: $\mathbf{FwBnR}(\Upsilon) = \bigcup_{n\in\mathbb{N}} F_n$ where $F_0 = \Upsilon$ and $F_{n+1} = \bigcup_{\tau\in\delta} \bigcup_{(q,Z,Z')\in F_n} \mathbf{FwBnR}(q, Z, Z', \tau)$.

On the other hand, for each set $\mathcal{Z} \subseteq Q \times \mathcal{Z}_{\mathcal{A}}$, we define $\mathbf{Cyc}(\mathcal{Z})$ as the subset of $\mathcal{Z}$ that contains only regions which lie on a cycle in $\mathsf{Reg}_{\mathcal{A}}$,

$$\mathbf{Cyc}(\mathcal{Z}) = \bigcup \big\{(q, Z) \subseteq \mathcal{Z} \mid \forall R \subseteq Z, R \in \mathcal{R}_{\mathcal{A}}, ((q, R), (q, R)) \in \delta^{+}_{\mathcal{R}}\big\}$$

Proposition 9. $\mathbf{Fwd}(\mathbf{Cyc}(\mathcal{Z})) = \mathbf{Fwd}\Big(\nu X.\big(\mathcal{Z} \cap \mathbf{Fwd}(X)\big)\Big)$.

This result is a corollary of a property related to strongly connected components (s.c.c.) in general graphs, that we give in the following:

Lemma 4. *Consider a finite graph $G = (V, E)$ ($E \subseteq V \times V$, $card(V) < \infty$) in which $E = E_1 \cup E_2$ with $E_1 \cap E_2 = \emptyset$ and such that E does not contain self loops. Denote E^*, E_1^* and E_2^* the reflexive-transitive closure of E, resp. E_1, E_2, and for any $U \subseteq V$, define as usual $E(U) = \big\{v \in V \mid \exists u \in U, (u, v) \in E\big\}$ and $E^*(U) = \mu X.(U \cup E(X))$. Also denote:*

$$\mathbf{Fwd}(U) = \big\{v \in V \mid \exists u \in U, v_1, v_2 \in V, (u, v_1), (v_2, v) \in E_2^*, (v_1, v_2) \in E_1\big\}$$
$$\mathsf{SCC}_{\geq 2}(U) = \bigcup \big\{W \subseteq U \mid W \text{ is a s.c.c. with } card(W) \geq 2\big\}$$

Suppose that $U = E^*(U)$ and there are no nontrivial cycles containing only edges from E_2. Then

$$E^*(\mathsf{SCC}_{\geq 2}(U)) = E^*\Big(\nu X.(U \cap \mathbf{Fwd}(X)\Big)$$

Proof. For the left-to-right inclusion, take $v \in \mathsf{SCC}_{\geq 2}(U)$, which means that there exists a s.c.c. $W \subseteq U$ with $card(W) \geq 2$ and $v \in W$. Note first that $U = E^*(U)$ implies $\mathbf{Fwd}(E^*(W)) \subseteq U$. On the other hand, due to the fact that E_2 contains no nontrivial cycles, we must have that $\mathbf{Fwd}(E^*(W)) = E^*(W)$. This means that $E^*(W)$ is a fixpoint for the mapping $X \mapsto U \cap \mathbf{Fwd}(X)$, and hence $v \in W \subseteq \nu X.(U \cap \mathbf{Fwd}(X))$ which is the greatest fixpoint.

For the reverse proof, take W with $W = U \cap \mathbf{Fwd}(W)$ and pick some $v \in W$. The fixpoint equation implies that for any k there exists a sequence of vertices $v_1, \ldots, v_k \in W$ such that $(v_i, v_{i+1}) \in E$ ($1 \leq i \leq k$) with $v_{k+1} = v$. Also $v_i \neq v_{i+1}$, since E contains no self loops. By finiteness of V, there exist $1 \leq j_1 < j_2 - 1 \leq k$ with $v_{j_1} = v_{j_2}$. It follows that there exists a s.c.c. $W' \subseteq W$ with $v_{j_1}, \ldots, v_{j_2-1} \in W'$ and also $card(W') \geq 2$. But this implies that $v \in E^*(W') \subseteq E^*(\mathsf{SCC}_{\geq 2}(U))$. $\square$

Remark 2. Note that, in general, $\mathsf{SCC}_{\geq 2}(U) \neq \nu X.\big(U \cap \mathbf{Fwd}(X)\big)$.

Proposition 9 is then an easy corollary of this lemma, if we take G as the region graph, with $E_1 = \xrightarrow{\mathtt{t}}$ and $E_2 = \xrightarrow{\downarrow}$. Note that our assumption on the given timed automaton ensure the hypotheses in the lemma. More specifically, the existence of the extra clock which is reset and is checked to be > 0 on each transition implies the hypothesis on E_2. not containing nontrivial cycles.

The construction of $\mathbf{Fwd}(\mathbf{Cyc}(\mathcal{Z}))$ involves the fixpoint computation of the inner greatest fixpoint, as the "limit" of the the sequence $C_0 = \mathcal{Z}$ and $C_{n+1} = C_n \cap \mathbf{Fwd}(C_n)$. Then, when this sequence stabilizes, we apply forward closure.

Our symbolic algorithm for computing $\mathsf{RegReach}_{\Delta\to 0}(q_0, \mathbf{0}_n)$ is the following:

Algorithm 1 Construction of $\mathsf{Reach}_{\mathcal{E}(\mathcal{A})}(q_0, \mathbf{0}_n, \mathbf{0}_n)$

$BReach := \{(q_0, \mathbf{0}_n, \mathbf{0}_n)\}$; $PrevBReach := \emptyset$;
while $BReach \neq PrevBReach$
  $PrevBReach := BReach$;
  $BReach := \mathbf{FwBnR}(BReach)$;
  $\mathcal{Z} := \{(q, Z') \mid \exists Z, (q, Z, Z') \in BReach\}$;
  $BReach := BReach \cup \{(q, Z, Z') \mid (q, Z') \in \mathbf{Fwd}(\mathbf{Cyc}(\mathcal{Z})), Z = \mathsf{t}\text{–}\mathsf{Nghbr}(Z')\}$;
end while ;
return $BReach$;

Theorem 2. *Let $\mathcal{A}$ be a bounded timed automaton with no self loops and in which there exists a clock x such that all transitions (q, C, X, q') have $x \in X$ and $C \wedge (x = 0)$ not satisfiable. Then $(q, R) \in \mathsf{RegReach}_{\Delta\to 0}(q_0, \mathbf{0}_n)$ if and only if, at the end of the above algorithm, there exists $(q, Z, Z') \in BReach$ such that $R \subseteq Z$.*

This result is a corollary of Proposition 8 and of Theorem 1. Note also that Corollary 1 is essential in the proof of this theorem, as it allows considering forward propagations of strongly connected components, instead of just cycles in the region graph.

Comparison with [DK06] The symbolic construction in the last algorithm is different from the one of [DK06]: in that paper, a *stable zone* W_σ is constructed, symbolically, *for each cycle* σ in the timed automaton. The fixpoint definition for W_σ is $W_\sigma = \nu X.(\mathbf{Fwd}(X) \cap \mathbf{Bck}(X))$. Hence, a preliminary analysis of the graph of the timed automaton is needed, in which all the cycles of the graph have to be constructed. It is well-known that the number of cycles in a graph is superexponential in the size of the graph, hence, at least in theory, the approach of [DK06] may lead to a superexponential time complexity.

One might also ask whether the approach from [DK06] may apply to a strongly connected component rather than to only one cycle at a time. The answer is negative in general, for the following reasons: first, as already [DK06] note, W_σ is in general larger than the set of regions which lie on a cycle in the region graph that is "induced" by σ. However, any region in W_σ is Δ-reachable (for any Δ) from any other region due to a convexity argument regulating the Δ-trajectories through the cycle σ. This argument no longer applies for strongly connected components with more than one cycle. As a counterexample, in the timed automaton in Figure 3, if we put $\mathcal{Z} = \{(q_2, R) \mid R \in \mathcal{R}_\mathcal{A}\}$, then

$$W = \mathcal{Z} \cap \nu X.(\mathbf{Fwd}(X) \cap \mathbf{Bck}(X)) = \{(q_2, (x \in [0,3]) \vee (x \in [4,6]))\}$$

which is not convex (here $\mathbf{Bck}$ is the backward propagation of a set of zones). And it should be clear that, from the state region $(q_2, x \in [4,5])$, no region within the $\mathcal{A}$-zone $(q_2, x \in [0,3])$ can be Δ-reached. More generally, reaching some region within W does not give guaranty that all W is Δ-reachable.

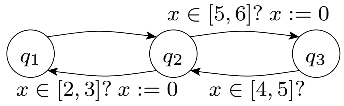

Fig. 3. An example of a non-convex generalization of "stable sets" of [DK06].

5 Conclusions

We have presented a construction for solving the following safe implementation problem: given a timed automaton $\mathcal{A}$ and some n-dimensional zone Z, does there exist a clock drift Δ for which no trajectory in $\mathcal{A}$ in which clocks drifts with at most Δ units reaches Z? The construction generalizes [Pur98], by allowing also the handling of non-closed constraints. We also give a symbolic algorithm that builds the set of zones that are reachable with arbitrarily small clock drifts.

Our algorithm works by constructing, symbolically, forward propagations of strongly connected components in the region graph. Most of the constructions in our algorithm work also with representations of *sets of* zones, like the clock decision diagrams of [LPWY99]. The only construction that could raise problems is $\mathsf{t}{-}\mathsf{Nghbr}$, which, as defined, can only be applied to one DBM at a time. We are interested in finding ways to bypass this problem for constructing an algorithm which is fully compatible with CDDs.

On the other hand, our technique of considering strongly connected components instead of just cycles in the region graph can be easily applied to automata containing only closed constraints, as in [Pur98,DK06]. It is possible that, in that setting, the above compatibility problem between $\mathsf{t}{-}\mathsf{Nghbr}$ and CDDs be solvable in a easier way, since in the closed constraints case, region neighborhoodness means regions with nonempty intersection.

Up to the author's knowledge, there exist no algorithms allowing the symbolic computation of the non-trivial strongly connected components in a graph, employing only the set-based constructions that are used in reachability algorithms for timed systems – that is, union, intersection, forward or backward propagation. Symbolic algorithms with good complexity like [GPP03,BGS00] use a "pick" function which returns a single node in the graph, and employ set difference. First, picking a region in a zone, though not an expensive operation, might prove to be a harmful operation w.r.t. set-based structures like clock-difference diagrams. Secondly, set difference, in our setting, amounts to DBM subtraction, which is known not to be a "nice" operation on DBMs. Some heuristics for DBM subtraction have been investigated in [DHLP06].

References

[AD94] R. Alur and D. Dill. A theory of timed automata. *Theoretical Computer Science*, 126:183–235, 1994.

[AHV93] R. Alur, T. A. Henzinger, and M. Vardi. Parametric real-time reasoning. In *Proceedings of STOC'93*, pages 592–601. ACM, 1993.

[AT05] K. Altisen and S. Tripakis. Implementation of timed automata: An issue of semantics or modeling? In *Proceedings of FORMATS'05*, volume 3829 of *LNCS*, pages 273–288. Springer Verlag, 2005.

[ATP01] R. Alur, S. La Torre, and G. J. Pappas. Optimal paths in weighted timed automata. In *Proceedings of HSCC'01*, volume 2034 of *LNCS*, pages 49–62. Springer Verlag, 2001.

[BDM⁺98] M. Bozga, C. Daws, O. Maler, A. Olivero, S. Tripakis, , and S. Yovine. Kronos: a model-checking tool for real-time systems. In *Proceedings of CAV'98*, volume 1427 of *LNCS*, pages 546–550, 1998.

[BGS00] R. Bloem, H.N. Gabow, and F. Somenzi. An algorithm for strongly connected component analysis in $n \log n$ symbolic steps. In *Proceedings of FMCAD'00*, volume 1954 of *LNCS*, pages 37–54, 2000.

[BLR05] P. Bouyer, Fr. Laroussinie, and P.-A. Reynier. Diagonal constraints in timed automata: Forward analysis of timed systems. In *Proceedings of FORMATS'07*, volume 3829 of *LNCS*, pages 112–126. Springer Verlag, 2005.

[DDMR04] M. DeWulf, L. Doyen, N. Markey, and J.F. Raskin. Robustness and implementability of timed automata. In *Proceedings of FORMATS/FTRTFT'04*, volume 3253 of *LNCS*, pages 118–133. Springer Verlag, 2004.

[DDR05a] M. DeWulf, L. Doyen, and J.-F. Raskin. Systematic implementation of real-time models. In *Proceedings of FM'05*, volume 3582 of *LNCS*, pages 139–156. Springer Verlag, 2005.

[DDR05b] M. DeWulf, L. Doyen, and J.F. Raskin. Almost asap semantics: from timed models to timed implementations. *Formal Aspects of Computing*, 17(3):319–341, 2005.

[DHLP06] A. David, J. Hakanson, K.G. Larsen, and P. Petersson. Model checking timed automata with priorities with DBM subtraction. In *Proceedings of FORMATS'06*, volume 4202 of *LNCS*, pages 128–142, 2006.

[Dim06] C. Dima. Dynamical properties of timed automata revisited. Technical Report TR-2006-03, LACL, Université Paris 12, 2006.

[DK06] C. Daws and P. Kordy. Symbolic robustness analysis of timed automata. In *Proceedings of Formats'06*, volume 4202 of *LNCS*, pages 143–155, 2006.

[GPP03] R. Gentilini, C. Piazza, and A. Policriti. Computing strongly connected components in a linear number of symbolic steps. In *Proceedings of SODA'03*, pages 573–582, 2003.

[HHWT97] T.A. Henzinger, P.-H. Ho, and H. Wong-Toi. HYTECH: A model checker for hybrid systems. *Software Tools for Technol. Transfer*, 1:110–122, 1997.

[HKPV98] T.A. Henzinger, P.W. Kopke, A. Puri, and P. Varaiya. What's decidable about hybrid automata. *J. Comput. Syst. Sci*, 57:94–124, 1998.

[LPWY99] K.G. Larsen, J. Pearson, C. Weise, and Wang Yi. Clock difference diagrams. *Nord. J. Comput.*, 6:271–298, 1999.

[LPY97] K. G. Larsen, Paul Petterson, and Wang Yi. Uppaal: Status & developments. In *Proceedings of CAV'97*, LNCS, pages 456–459, 1997.

[Pur98] A. Puri. Dynamical properties of timed automata. In *Proceedings of FTRTFT'98*, volume 1486 of *LNCS*, pages 210–227. Springer Verlag, 1998.

[WT97] H. Wong-Toi. Analysis of slope-parametric rectangular automata. In *Hybrid Systems*, volume 1567 of *LNCS*, pages 390–413. Springer Verlag, 1997.

[Yov98] S. Yovine. Model-checking timed automata. In *Lectures on Embedded Systems*, volume 1494 of *LNCS*, pages 114–152, 1998.