Timed Automata with Periodic Clock Constraints [1]

Christian Choffrut
L.I.A.F.A., Université Paris 7, Tour 55-56, 1 er étage,
2 Pl. Jussieu – 75 251 Paris Cedex – France
Christian.Choffrut@liafa.jussieu.fr

Massimiliano Goldwurm
Dipartimento di Scienze dell'Informazione, Università degli Studi di Milano
via Comelico, 39 – 20135 Milano – Italy
goldwurm@dsi.unimi.it

June 1999

Abstract

The traditional constraints on the clocks of a timed automaton are based on real intervals, e. g., the value of a clock belongs to the interval $(0, 1)$. Here, we introduce a new set of constraints, which we call "periodic", and which are based on regularly repeated real intervals, e. g., the value modulo 2 of a clock belongs to the interval $(0, 1)$ which means that it belongs to $(0, 1)$ or $(2, 3)$ or $(4, 5)$ ....

Automata with these new constraints have greater expressive power than the automata with traditional sets while satisfiability remains decidable. We address questions concerning ϵ-moves and determinism: simulation of automata with periodic constraints by automata with traditional constraints, properties of deterministic automata with periodic constraints (like closure under Boolean operations and decidability of the inclusion problem) and removal of ϵ-moves under certain conditions. Then, we enrich our model by introducing "count-down" clocks and show that the expressive power is not increased. Finally, we study three special cases: 1) all transitions reset clocks, 2) no transition reset clocks, and 3) the time domain is discrete and prove the decidability of the inclusion problem under each of these hypotheses.

Keywords: model checking, analysis of real-time systems, timed automata, ω-automata.

[1] This work has been supported by MURST Research Program "Unconventional computational models: syntactic and combinatorial methods".

1 Introduction

Emphasis on concrete time, i. e., on *when* events occur and not only in *which order* they occur, is a vivid concern of the ongoing research on how real-time systems should be modelled or verified. Among the most popular models to be found in the literature are different kinds of real-time temporal logics that are extensions of the classical time or branching temporal logics [6, 1, 19, 8, 9], and different types of timed finite automata obtained by providing traditional automata with clocks controlling the triggering of transitions [3, 5, 7]. A third approach is based on fragments of monadic second order logics by resorting to a restricted use of distance between two elements on the real line [8, 7, 4, 22]. Under some hypotheses, the expressive equivalence of these models can be established. The great merit of these extensions of "untimed" notions is that they are very powerful while maintainig decidable many interesting questions.

The present paper deals more specifically with the "automata" model as exposed in [2, 3] for example. More precisely, a fixed set of clocks is given along with an automaton. These clocks share the same unit of time but can be reset independently when a transition is traversed. A transition is enabled provided certain conditions on the values of the clocks are met. This initial model has been extended in various directions to accomodate precision of computation [15], ϵ-transitions (also called *silent* transitions, ϵ-moves or *silent* moves) [11, 14, 18], convergent sequences of time [12], hybrid models [16], etc. .

The original model of timed automata is specified in terms of conditions involving intervals only: e. g., a transition is enabled when $1 < x < 2$ and $y > 5$ holds. In a certain sense it allows counting up to or from a certain threshold value. It also allows conditions of the previous form where the difference of two clocks is substituted for one single clock, but it is well known that the expressive power does not increase. On the other hand, with more general conditions such as $x - 2y > 3$, the emptiness problem for languages recognized by timed automata becomes undecidable. The notion of periodic clock constraints stems from the theory of rational relations on $\mathbb{N}$. Here, we introduce a family of relations that extend the expressive power of the traditional constraints while keeping the emptiness problem decidable. This is done by allowing "modulo" counting primitives, e. g. by considering for a given clock x the values $2, 5, \ldots, 3k+2, \ldots$. For this reason we call the constraints "periodic" as opposed to the traditional "aperiodic" ones. This is a natural extension since, regardless of the previous theoretical reason, many processes in different fields (physical, biologic, social, etc.) have a periodic behaviour. To our knowledge this formalism was mentioned in [8] only for the discrete time [2]. When applied to traditional timed automata the ability of using periodic constraints reduces significantly the number of transitions needed to produce a given behaviour.

The main results of this work are the following. We start by comparing the expressive powers of timed automata using periodic and aperiodic constraints. We give a construction that transforms automata with periodic clock constraints into equivalent automata with aperiodic clock constraints at the cost of introducing some possible ϵ-transitions. In fact the expressive power of the family of automata with periodic constraints and no ϵ-transitions lies strictly between the family of automata with aperiodic constraints

[2] Independently, Demichelis and Zielonka have recently studied a more general timed model of computation called "control timed automaton" which admits transitions with periodic clock constraints as a special case [13].

and no ϵ-transitions and the family of automata with aperiodic constraints and possible ϵ-transitions. Similar properties hold for deterministic timed automata with periodic constraints. Moreover, for this model, the inclusion problem is decidable and the corresponding class of languages is closed under the Boolean operations.

Next we tackle the problem of removing ϵ-transitions. For finite (whether one or multi-tape) untimed automata, such moves can be eliminated easily. For timed automata this no longer holds. It was proved in [11] that ϵ-transitions without resets can be removed in automata using aperiodic constraints. The same result holds for automata with the new set of constraints. With traditional constraints, further ϵ-transitions may be eliminated, to wit those that do not lie on a loop [14]. This no longer holds in our case for a reason that is closely related to some Zeno property.

Dually to clocks that measure the time elapsed since an event occurred, we may consider "count-down" clocks measuring the time left before an event occurs. Many examples of some type of count-down clocks can be found in the literature, whether in the framework of automata or temporal logics [5, 7, 22]. We show that for automata with periodic clock constraints, "count-down" clocks can be eliminated without introducing new ϵ-moves.

In section 7 we investigate two special cases, reset-free and pure reset automata with periodic constraints. Under these particular hypotheses a stronger result holds since inclusion of languages recognized by timed automata is decidable. We also show that when the domain is the set of integers (with possible ϵ-transitions and periodic clock constraints), all clocks can be replaced by a unique pure reset clock, which extends some results of [8] and [17].

2 Preliminaries

We first assume the set $\mathbb{R}_+$ of nonnegative reals as time domain. In section 7.2 we study timed automata under the assumption of a discrete time domain. So, we define a *time sequence* as a finite or an infinite sequence

$$\{t_i\}_i = t_1, t_2, \ldots, t_i, \ldots$$

where $t_i \in \mathbb{R}_+$ and $t_i < t_{i+1}$ for every index i, which is divergent whenever it consists of infinitely many terms (i.e. $\lim_{i\to+\infty} t_i = +\infty$).

Now, given a finite alphabet Σ, a *timed string* (also called *timed word*) is a finite or infinite sequence of the form

$$\{(\sigma_i, t_i)\}_i = (\sigma_1, t_1)(\sigma_2, t_2)\cdots(\sigma_i, t_i)\cdots$$

where $\sigma_i \in \Sigma$ for every i and $\{t_i\}_i$ is a time sequence. We denote by $(\Sigma \times \mathbb{R}_+)^\infty$ the set of all such sequences and hence a *timed language* is defined as a subset of $(\Sigma \times \mathbb{R}_+)^\infty$.

In order to specify timed languages, the standard notion of finite automaton is modified by introducing enabling times for transitions. These times are controlled by predicates which determine when the transition may be executed. There is no universal time of reference given with the automaton, only a fixed number of clocks sharing the same unit of time but that can be reset independently.

We start by giving a general definition of clock constraints. First, recall that a *clock* is a variable with value in $\mathbb{R}_+$.

DEFINITION 1 *Given a sequence X of n clocks, a set $\Phi(X)$ of* clock constraints *is a family of subsets of $\mathbb{R}_+^n$ that is a closed under the Boolean set operations.*

Hence we may view every element of $\Phi(X)$ as a set of values for the clocks in X. Therefore, we often denote a clock constraint by a condition or a predicate concerning one or more clocks.

Now, let us give the usual definition of an automaton that recognizes finite and infinite timed strings at once.

DEFINITION 2 *A* timed (Büchi) automaton *on a set of clock constraints $\Phi(X)$ is a tuple $\mathcal{A} = (\Sigma, Q, I, F, R, X, T)$, where Σ is a finite set of events, Q is a finite set of states, I, F and R are subsets of Q (i.e. the set of initial, final and repeated states, respectively), X is the set of clocks and $T \subseteq Q \times \Phi(X) \times (\Sigma \cup \{\epsilon\}) \times 2^X \times Q$ is a finite set of transitions (or moves).*

Then, let us define the timed words accepted by $\mathcal{A}$. First, we recall that an *assignment* of the clocks is a function of $\nu : X \to \mathbb{R}_+$. A finite (resp. infinite) timed word

$$(\sigma_1, t_1)(\sigma_2, t_2) \cdots (\sigma_i, t_i) \cdots \in (\Sigma \cup \{\epsilon\} \times \mathbb{R}_+)^\infty$$

is *accepted* (or recognized) by $\mathcal{A}$ if there exist a finite (resp. infinite) sequence of assignments ν_i of the clocks, $i = 0, 1, 2, \ldots$, with $\nu_0(x) = 0$ for every x, and a finite (resp. infinite) sequence of states q_i, $i = 0, 1, 2, \ldots$, with $q_0 \in I$, ending in F (resp. visiting R infinitely many times), such that, for each index i, there is a transition $(q_{i-1}, \phi_i, \sigma_i, X_i, q_i) \in T$ with the following properties:

1. each value $\nu_{i-1}(x) + t_i - t_{i-1}$ satisfies the constraints ϕ_i (with the initial condition $t_0 = 0$),
2. for every $x \in X$ we have

$$\nu_i(x) = \begin{cases} 0 & \text{if } x \in X_i, \\ \nu_{i-1}(x) + t_i - t_{i-1} & \text{otherwise.} \end{cases}$$

The sequence $\{(q_i, \nu_i)\}_{i \geq 0}$ is called a *run* of the automaton over the timed string $\{(\sigma_i, t_i)\}_{i \geq 1}$. Each element (q_i, ν_i) is a so-called *extended state* of the timed automaton.

Whenever ϵ is the third component of a transition $\tau \in T$, we say that τ is an *ϵ-transition* or a *silent move*. Since silent moves are usually ignored to describe the behaviour of a system, extending the previous definition, we say that a timed string $\gamma \in (\Sigma \times \mathbb{R}_+)^\infty$ is accepted (or recognized) by $\mathcal{A}$ if the automaton accepts a word $\gamma' = \{(\sigma_i, t_i)\}_i \in (\Sigma \cup \{\epsilon\} \times \mathbb{R}_+)^\infty$ and γ is obtained from γ' by simply erasing all pairs (ϵ, t_i). The runs of $\mathcal{A}$ over γ are the runs over γ'.

The *language accepted* by the timed automaton $\mathcal{A}$ is the set of all the timed words in $(\Sigma \times \mathbb{R}_+)^\infty$ accepted by $\mathcal{A}$. Two timed automata are *equivalent* if they accept the same language.

We say that our timed Büchi automaton $\mathcal{A}$ is *restricted* if every silent move in $\mathcal{A}$ does not reset any clock. Moreover, $\mathcal{A}$ is *deterministic* if T does not contain any ϵ-transition, I consists of one state only and, for every pair of distinct moves

$$(q, \phi_1, \sigma, X_1, p_1), (q, \phi_2, \sigma, X_2, p_2) \in T$$

with the same starting state $q \in Q$ and the same label $\sigma \in \Sigma$, the equality $\phi_1 \cap \phi_2 = \emptyset$ holds [3].

As usual, we describe timed automata by labelled directed graphs: states are represented by nodes and transitions by labelled arrows of the form $q \xrightarrow{\phi;\sigma;Y} p$ with obvious meaning; moreover, repeated, final and initial states are represented by a double circle, an outcoming and an incoming arrow, respectively.

The traditional notion of timed automata (with ϵ-transitions) corresponds to our definition when $\Phi(X)$ is defined by the grammar

$$\phi := x \leq a \mid x \geq a \mid x - y \leq a \mid x - y \geq a \mid \neg\phi \mid \phi \wedge \phi,$$

where $x, y \in X$, $a \in \mathbb{Q}_+$ (the set of nonnegative rational numbers). In order to distinguish these constraints from the periodic ones (introduced in the next section), we call them *traditional* or *aperiodic* constraints. We denote by $\mathcal{T}_\epsilon^a$ the family of all such automata and by $\mathcal{T}^a$ the subset of all $\mathcal{A} \in \mathcal{T}_\epsilon^a$ that have no ϵ-transitions. Analogously, $\mathcal{L}_\epsilon^a$ and $\mathcal{L}^a$ denote the classes of timed languages recognized by automata in $\mathcal{T}_\epsilon^a$ and $\mathcal{T}^a$, respectively.

3 New clock constraints

Now we define the family of clock constraints whose study is the main purpose of this paper.

For every subset $H \subseteq \mathbb{R}$ and every $\lambda \in \mathbb{R}_+$, we denote by $T_\lambda(H)$ the set

$$T_\lambda(H) = \{h + k\lambda \mid h \in H, k \in \mathbb{N}\}$$

By an *interval* of $\mathbb{R}_+$ we mean an open (resp. left semi-open, right semi-open, closed) interval of the form (a, b) (resp. $(a, b]$, $[a, b)$, $[a, b]$) with $a, b \in \mathbb{R}_+$, $a \leq b$. The interval is *rational* (resp. *integer*) if further $a, b \in \mathbb{Q}_+$ (resp. $a, b \in \mathbb{N}$). For simplicity, $T_\lambda[a, b]$ and $T_\lambda[a]$ stand for $T_\lambda(H)$ when $H = [a, b]$ and $H = [a, a]$, respectively, and we use the same simplified notation for open and semi-open intervals.

DEFINITION **3** *Given a family X of clocks, the set $\Phi(X)$ of* periodic clock constraints *is defined by the grammar*

$$\phi := x \in T_\lambda[a, b] \mid x - y \in T_\lambda[a, b] \mid \neg\phi \mid \phi \wedge \phi \tag{1}$$

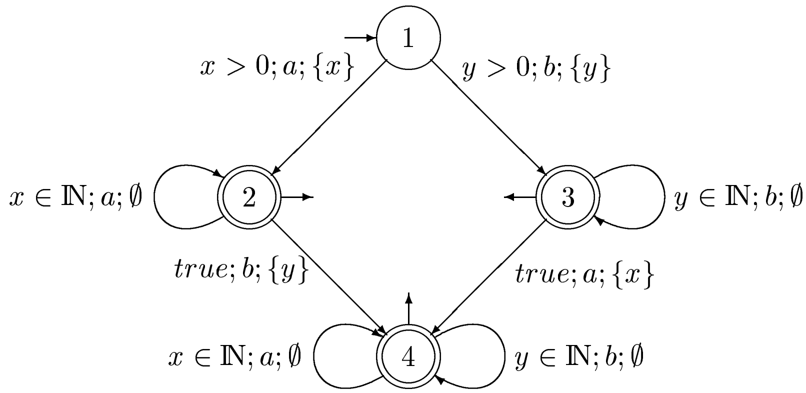
where $a, b, \lambda \in \mathbb{Q}_+$ and $x, y \in X$. The constraints of the form $x \in T_\lambda[a, b]$ or $x - y \in T_\lambda[a, b]$ are called atoms *and a constraint involving atoms with two clocks is called* diagonal.

Clearly any $\phi \in \Phi(X)$, together with the implicit conditions $x \geq 0$ for $x \in X$, defines a subset of $\mathbb{R}_+^n$.

EXAMPLE **3.1** Let $\Sigma = \{a, b\}$ and consider the language of all (nonempty) timed strings $\{(\sigma_i, t_i)\}_i \in (\Sigma \times \mathbb{R}_+)^\infty$ such that, if t_u and t_v are the times of the first occurrence of a and b respectively, then both t_u and t_v are greater than 0 and the following property holds for every index i of the sequence:

$$\sigma_i = a \Rightarrow t_i - t_u \in \mathbb{N} \wedge \sigma_i = b \Rightarrow t_i - t_v \in \mathbb{N}$$

Such a language is recognized by the timed automaton with periodic constraints described in the following picture, where the conditions $x > 0$, *true* and $x \in \mathbb{N}$ are represented by the periodic predicates $x \in T_1(0,1]$, $x \in T_1[0,1)$ and $x \in T_1[0]$, respectively.



□

In the following, we denote by $\mathcal{T}^p_\epsilon$ the set of all timed automata on periodic clock constraints and by $\mathcal{T}^p$ the subset of all $\mathcal{A} \in \mathcal{T}^p_\epsilon$ that have no silent move. Similarly, $\mathcal{L}^p_\epsilon$ ($\mathcal{L}^a$) denotes the class of all languages accepted by timed automata in $\mathcal{T}^p_\epsilon$ ($\mathcal{T}^a$, resp.).

It is easy to see that each traditional constraint is also a periodic constraint. For instance, the constraint $x \geq a$ is expressed in our language by $x \in T_\lambda[a, b]$ with $\lambda = b - a$. This implies $\mathcal{L}^a_\epsilon \subseteq \mathcal{L}^p_\epsilon$ and $\mathcal{L}^a \subseteq \mathcal{L}^p$.

On the other hand, we prove in Section 4 that every timed automaton with periodic clock constraints is equivalent to a timed automaton with aperiodic ones. The proof is based on a sort of canonical form for timed automata we present in the following section.

3.1 Periodic constraints in canonical form

We now present some properties that allow to simplify the form of the timed automata with periodic clock constraints leaving unchanged the expressive power of the model.

We first consider the removal of diagonal constraints. Indeed, a folkloric result states that in the aperiodic case diagonal constraints can be eliminated. The proof of this property (see, for instance [10]) does not depend on the particular set of constraints assumed for the automaton and hence carries over word for word by substituting the traditional constraints with the periodic constraints.

PROPOSITION **1** *Every timed automaton $\mathcal{A} \in \mathcal{T}^p_\epsilon$ admits an equivalent timed automaton $\mathcal{B} \in \mathcal{T}^p_\epsilon$ having no diagonal constraint.*

A further simplification is given by the usual restriction to timed automata with integer constraints [3]. Indeed, given a timed automaton $\mathcal{A}$, we can multiply all the constants occurring in the constraints by the least common multiple m of their denominators. Hence, we obtain a new timed automaton $\mathcal{A}'$ with integer constraints that is "essentially" equivalent to the previous one, i.e. $\mathcal{A}$ recognizes the timed sequence $\{(\sigma_i, t_i)\}_{i\geq 1}$ if and only if $\mathcal{A}'$ accepts the timed sequence $\{(\sigma_i, mt_i)\}_{i\geq 1}$. In the following we say that a periodic constraint is *integer* if its atoms are of the form $x \in T_\lambda(H)$ or $x - y \in T_\lambda(H)$, where $\lambda \in \mathbb{N}$ and H is an integer interval.

However, the most significant simplification is due to the following two propositions whose proof is given in the Appendix A. Observe that, strictly speaking, this does not yield a canonical form, just a sort of simplified form that can be useful in proving the properties of our model.

LEMMA **3.1** *For every integer periodic constraint ϕ there exists $m \in \mathbb{N}$ such that*

$$\phi = \bigvee_{i \in I} \bigwedge_{j \in J_i} B_{ij} \tag{2}$$

where I and each J_i are finite sets of indices and every B_{ij} is either an atom of type

$$i) x = i, \quad ii) x \in (i, i+1), \quad iii) x \in T_m[j], \quad iv) x \in T_m(j, j+1) \tag{3}$$

where $i, j \in \mathbb{N}$, $0 \leq i < m$ and $m \leq j < 2m$, or a diagonal atom of the form

$$v) x - y = i, \quad vi) x - y \in (i, i+1), \quad vii) x - y \in T_m[j], \quad viii) x - y \in T_m(j, j+1) \tag{4}$$

where i, j are defined as before.

THEOREM **1** *Let $\mathcal{A} \in \mathcal{T}_\epsilon^p$ be a timed automaton with integer constraints. Then, there exists an integer $m \in \mathbb{N}$ and a timed automaton $\mathcal{A}' \in \mathcal{T}_\epsilon^p$, equivalent to $\mathcal{A}$, such that each constraint in $\mathcal{A}'$ is a conjunction of atoms of the form (3) where i and j are integer, $0 \leq i < m$ and $m \leq j < 2m$. Furthermore, $\mathcal{A}'$ is deterministic if also $\mathcal{A}$ is.*

At last, we recall some general properties of timed automata without silent moves that reset clocks.

PROPOSITION **2** *For every restricted automaton $\mathcal{A} \in \mathcal{T}_\epsilon^p$ there exists an equivalent restricted $\mathcal{A}' \in \mathcal{T}_\epsilon^p$ such that every finite word accepted by $\mathcal{A}'$ is recognized through a finite run.*

PROPOSITION **3** *For every restricted timed automaton $\mathcal{A} \in \mathcal{T}_\epsilon^p$ there exists a restricted $\mathcal{A}' \in \mathcal{T}_\epsilon^p$, equivalent to $\mathcal{A}$, that does not contain any ϵ-cycle.*

These statements are proved in [11, 10] for timed automata with traditional constraints. However, the same proofs with obvious changes (and some simplifications due to our definitions) hold for periodic constraints as well.

4 Simulation of periodic clock constraints

We apply the previous results for establishing the equivalence of periodic and aperiodic clock constraints (when ϵ-transitions are allowed).

THEOREM **2** *Every timed automaton $\mathcal{A} \in \mathcal{T}_\epsilon^p$ admits an equivalent $\mathcal{A}' \in \mathcal{T}_\epsilon^a$.*

Proof. Let $\mathcal{A} \in \mathcal{T}_\epsilon^p$ be defined by the tuple $(\Sigma, Q, I, F, R, X, T)$. We assume $\mathcal{A}$ in a simplified form for which, however, we do not need the full power of Theorem 1. The following conditions are less restrictive and can be easily derived from the same theorem: we assume there exists $m \in \mathbb{N}$ such that every constraint in $\mathcal{A}$ is the Boolean connection of atoms of the following forms:

$$x \in H \ , \quad x \geq c \ , \quad x \in T_m(K) \ ,$$

where H and K are integer intervals, $c \in \mathbb{N}$ and $K \subseteq (0, m]$. Then, we define the new automaton $\mathcal{A}'$ by the following construction.

The idea is to associate with every clock x a twin clock $\overline{x}$ which records the value of x modulo m. In other words, for any run of $\mathcal{A}'$, denoting by $x(t)$ the value of the clock x at time $t \in \mathbb{R}_+$, all pairs of twin clocks x and $\overline{x}$ satisfy the relation $\overline{x}(t) = x(t) - \lfloor \frac{x(t)}{m} \rfloor m$ whenever $x(t)$ is not a multiple of m. Moreover, each clock $\overline{x}$ is reset whenever reaches the value m, so that $\overline{x}(t) \in [0, m]$ for every $t \in \mathbb{R}_+$.

This notation is extended to subsets by setting $\bar{Y} = \{\bar{y} \mid y \in Y\}$ for every $Y \subseteq X$. Moreover, for all subsets $Z \subseteq X$, we denote by $\psi(Z)$ the predicate $\bigwedge_{x \in Z} \bar{x} = m \wedge \bigwedge_{x \in Z^c} \bar{x} < m$. Similarly, for every constraint ϕ in $\mathcal{A}$, let $\bar{\phi}$ be obtained from ϕ by replacing each atom of the form $x \in T_m(K)$ by the condition $\overline{x} \in K$. Note that $\bar{\phi}$ leaves unchanged the possible aperiodic atoms of ϕ.

Then, T' is defined as the set of moves

$$q \xrightarrow{\bar{\phi} \wedge \psi(Z);\ \sigma;\ Y \cup \bar{Y} \cup \bar{Z}} p \tag{5}$$

and

$$q \xrightarrow{\psi(U);\ \epsilon;\ \bar{U}} q \tag{6}$$

where $q, p \in Q$, $q \xrightarrow{\phi, \sigma, Y} p \in T$, Z and U are subsets of X and $U \neq \emptyset$.

If $R \subseteq F$, the new automaton is $\mathcal{A}' = (\Sigma, Q, I, F, R, X \cup \bar{X}, T')$. If $R \backslash F \neq \emptyset$, the previous construction must be modified since all finite timed strings leading to a repeated state are accepted in $\mathcal{A}'$ by executing infinitely many ϵ-loops of the form (6). To avoid this, for each $q \in R \backslash F$, we define a new state q' simulating q and move from q to q' all loops of the form (6). Formally, let Q' be the set Q enriched as above and, for each $q \in R \backslash F$, cancel from T' all moves of the form (6) and add the transitions

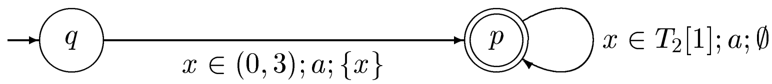
$$q \xrightarrow{\psi(U); \epsilon; \bar{U}} q', \quad q' \xrightarrow{\psi(U); \epsilon; \bar{U}} q', \quad q' \xrightarrow{\bar{\phi} \wedge \psi(Z);\ \sigma;\ Y \cup \bar{Y} \cup \bar{Z}} p$$

for every $q \xrightarrow{\phi; \sigma; Y} p \in T$ and every $Z, U \subseteq X$, $U \neq \emptyset$. Thus, if T'' is the new set of moves, $\mathcal{A}'$ is given by $\mathcal{A}' = (\Sigma, Q', I, F, R, X \cup \bar{X}, T'')$.

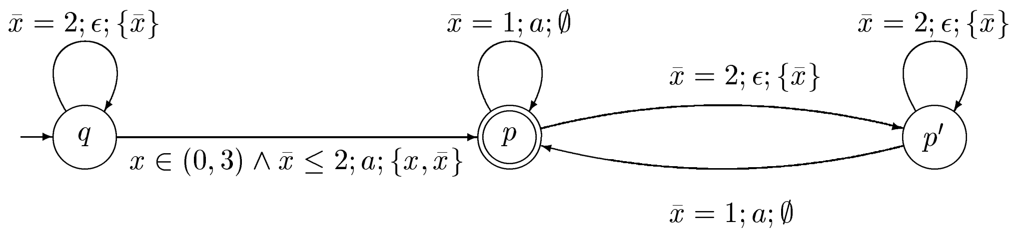
Now, reasoning by induction on the length of the timed words, it can be proved that $\mathcal{A}$ and $\mathcal{A}'$ recognize the same subset of $(\Sigma \times \mathbb{R}_+)^\infty$.

■

EXAMPLE **4.1** Consider the language of all infinite timed strings $\{(a, t_i)\}_{i \geq 1}$ such that $t_1 \in (0, 3)$ and, for every $i \geq 2$, $t_i - t_1$ is odd. This timed language is recognized by the following automaton with periodic constraints:



In this case, the construction of Theorem 2 defines the following timed automaton with traditional clock constraints (here $m = 2$):



A further example is given by Figure 1 where an aperiodic automaton is described that simulates the timed automaton of Example 3.1. This shows how the construction increases the number of transitions of the periodic automaton. We believe that the automaton of Figure 1 cannot be further reduced, i.e. no smaller automaton on traditional clock constraints can recognize the same language. □

COROLLARY **4** *The classes $\mathcal{L}^a_\epsilon$ and $\mathcal{L}^p_\epsilon$ coincide.*

As a consequence, timed automata with periodic clock constraints inherit the main properties of traditional timed automata [2, 3].

COROLLARY **5**

1) The emptyness problem for timed languages recognized by automata in $\mathcal{T}^p_\epsilon$ is decidable;

2) The universality problem for timed languages recognized by automata in $\mathcal{T}^p$ is undecidable;

3) Both classes $\mathcal{L}^p_\epsilon$ and $\mathcal{L}^p$ are closed under union, intersection but not under the complement.

4.1 Timed automata without silent moves

When no ϵ-transition is allowed, the new clock constraints are strictly more expressive than the traditional ones. For instance, it is known that the timed languages $Delay(k,p)$ ($k, p \in \mathbb{Q}_+$, $0 \leq k \leq p$), $Even$ and Int, given by

$$\begin{aligned} Delay(k,p) &= \{\{(\sigma_i, t_i)\}_{i \geq 1} \in (\Sigma \times \mathbb{R}_+)^\infty \mid \forall i \geq 1 \;\; t_{i+1} - t_i = k \bmod p\} \\ Even &= \{\{(\sigma_i, t_i)\}_{i \geq 1} \in (\Sigma \times \mathbb{R}_+)^\infty \mid \forall i \geq 1 \;\; t_i \in 2\mathbb{N}\} \\ Int &= \{(\sigma_i, t_i)\}_{i \geq 1} \in (\Sigma \times \mathbb{R}_+)^\infty \mid \forall i \geq 1 \;\; t_i \in \mathbb{N}\}, \end{aligned}$$

do not belong to $\mathcal{L}^a$ [11]. However, it is easy to see that they lie in $\mathcal{L}^p$ and this proves the following

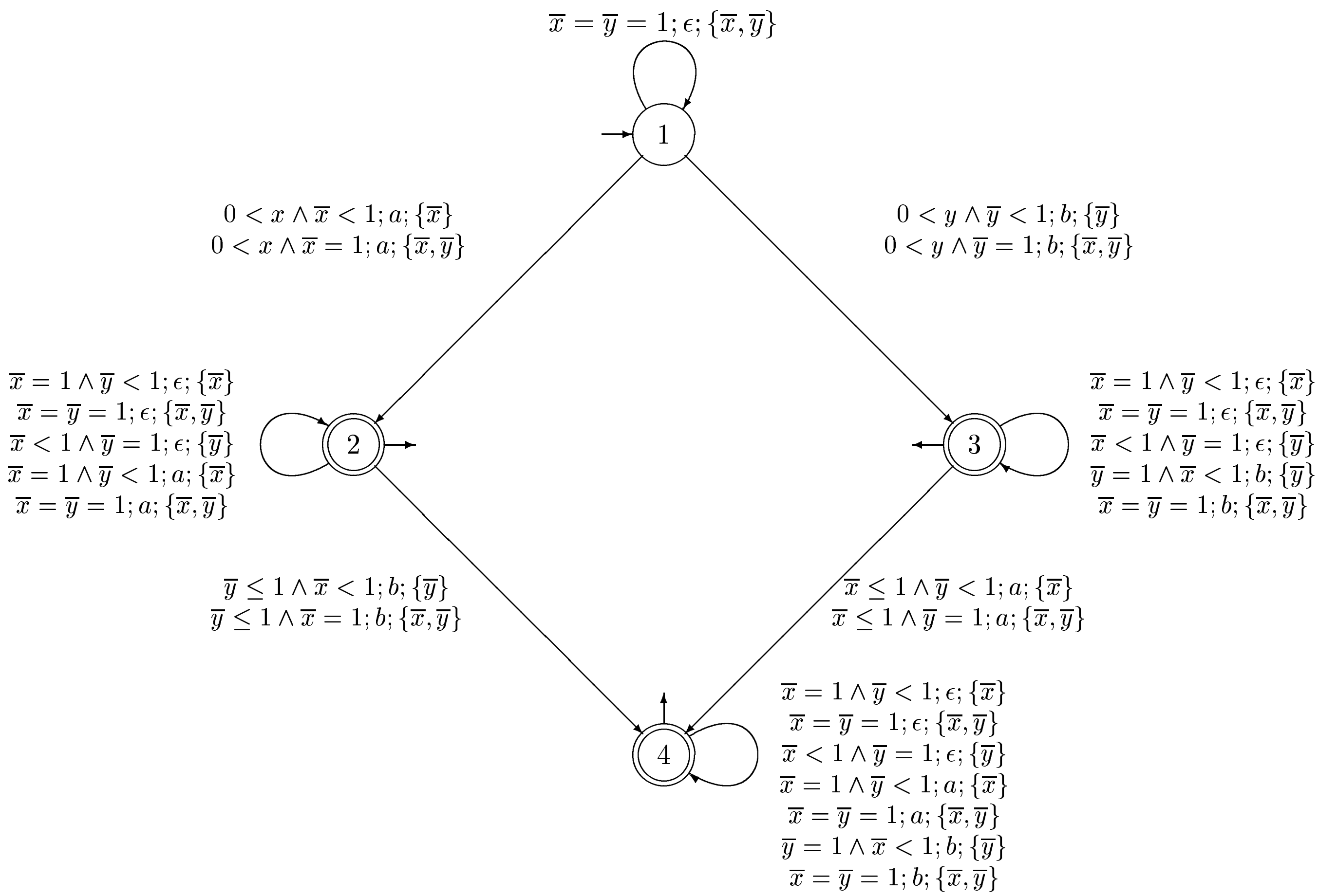
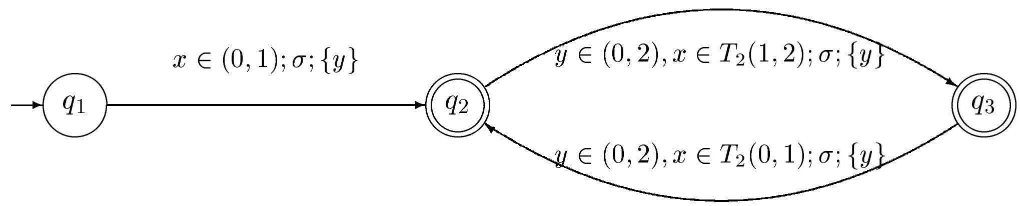


Figure 1: Aperiodic simulation of timed automaton in Example 3.1

Proposition **6** *The class $\mathcal{L}^a$ is strictly included in $\mathcal{L}^p$.*

Another significant example (where times of strings are not bounded to a discrete set) is the following.

Example **4.2** The set of all (infinite) timed strings $\{(\sigma, t_i)\}_{i\geq 1}$, such that $i-1 < t_i < i$ for all $i \geq 1$, does not belong to $\mathcal{L}^a$ [18]. However, the same language is accepted by the ϵ-free automaton on periodic clock constraints described by the following picture:



□

A timed language similar to the previous one allows to show that ϵ-transitions increase the expressive power of timed automata with periodic clock constraints. This extends an analogous property of traditional timed automata [11, 18].

Proposition **7** *The class $\mathcal{L}^p$ is strictly included in $\mathcal{L}^p_\epsilon$.*

Proof. Let us consider the set of all (infinite) timed strings $\{(\sigma, t_i)\}_{i\geq 1}$ such that there exists $\alpha \in \mathbb{R}_+$ satisfying the condition $\alpha + i - 1 < t_i < \alpha + i$ for every $i \geq 1$. An automaton with periodic constraints recognizing this set can be obtained from the picture in Example 4.2 by transforming q_1 into an ordinary state and adding a new initial state q_0 with the ϵ-transition

$$q_0 \xrightarrow{x > 0;\ \epsilon;\ \{x,y\}} q_1$$

This language is no longer recognizable by any timed automaton on periodic constraints without ϵ-transitions. Indeed, assume to the contrary that this is the case and let δ be the inverse of the least common multiple of the denominators of the coefficients involved in the constraints. By Proposition 1 we may assume that there is no diagonal constraint in the automaton. Then, any clock predicate occurring in the automaton has a constant value over the open intervals of the form $\prod_{1\leq i\leq m} (k_i\delta, (k_i+1)\delta)$, where $k_i \in \mathbb{N}$ for each i (and m is the number of clocks of the automaton).

Now, let $k > 0$ and consider the sequences $\{t_n\}$ and $\{t'_n\}$ defined by $t_1 = t'_1 = k$, $t_n = k + n - \frac{\delta}{2} - \frac{\delta}{2^{n+1}}$ and $t'_n = k + n - \frac{\delta}{2^{n+1}}$ for all $n > 1$. Then, we have $t_n, t'_n \in (k+n-\delta, k+n)$ and $t_n - t_1, t'_n - t'_1 \in (n-\delta, n)$ for all $n > 1$, while, for all $1 < j < n$, we have $t_n - t_j, t'_n - t'_j \in (n-j, n-j+\delta)$. By the form of the constraints this means that the behaviour of the automaton over the two sequences is the same. However, the timed sequence $\{(\sigma, t_n)\}_{n\geq 1}$ belongs to the language while $\{(\sigma, t'_n)\}_{n\geq 1}$ does not. ■

4.2 Deterministic timed automata

In this subsection we consider deterministic timed automata with periodic clock constraints. Their properties follows from Corollary 5 by reasoning as in the aperiodic case [3]. Here, we have to extend the notion of timed automaton by considering Muller acceptance conditions.

A *Muller* timed automaton is a tuple $\mathcal{A} = (\Sigma, Q, I, F, R, X, T)$ defined as a Büchi timed automaton except that R is a family of subsets of Q and an infinite word $\gamma \in (\Sigma \times \mathbb{R}_+)^\infty$ is accepted by $\mathcal{A}$ if there is a run r over γ such that the set of states entered by r infinitely often belongs to R. Applying the usual simulation of Muller acceptance condition by Büchi automata one can prove that the two timed automaton models (on a given set of clock constraints) define the same class of languages. However, this is not true if we consider deterministic timed automata as defined in Section 2. Actually, it can be proved that (both in the periodic and aperiodic case) deterministic timed Muller automata are more expressive than deterministic timed Büchi automata [3]. For this reason we assume timed Muller automata as model for deterministic computation in our context.

Let $\mathcal{DMT}^p$ and $\mathcal{DMT}^a$ be the class of deterministic timed Muller automata with periodic and aperiodic constraints, respectively, and let $\mathcal{DML}^p$ and $\mathcal{DML}^a$ be the corresponding classes of timed languages. Clearly, we have $\mathcal{DML}^a \subseteq \mathcal{L}^a$, $\mathcal{DML}^p \subseteq \mathcal{L}^p$ and $\mathcal{DML}^a \subseteq \mathcal{DML}^p$; moreover, the following properties can be proved.

PROPOSITION **8**

1) Every timed automaton $\mathcal{A} \in \mathcal{DMT}^p$ admits at most one run over any timed word;

2) The class $\mathcal{DML}^a$ is strictly included in $\mathcal{DML}^p$;

3) The class $\mathcal{DML}^p$ is closed under union, intersection and complement;

4) The class $\mathcal{DML}^p$ is strictly included in $\mathcal{L}^p$;

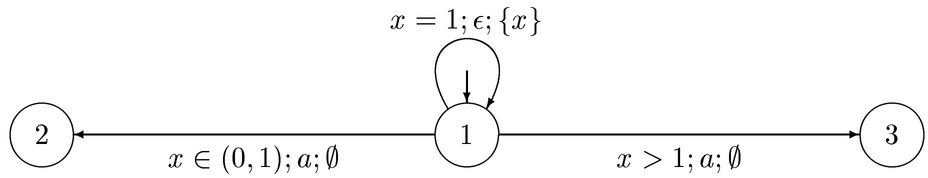
5) It is decidable, for two timed automata $\mathcal{A}_1 \in \mathcal{T}^p_\epsilon$ and $\mathcal{A}_2 \in \mathcal{DMT}^p$, whether the timed language L_1 recognized by $\mathcal{A}_1$ is included in the timed language L_2 recognized by $\mathcal{A}_2$.

Proof. The first statement follows by observing that, for every extended state in a run of $\mathcal{A}$, the next transition is uniquely determined by the next input symbol with the corresponding occurrence time. Statement 2) is a consequence of Example 4.2 that describes a deterministic timed automaton (which is a Muller automaton with family of repeated states $R = \{\{q_2, q_3\}\}$) recognizing a timed language not included in $\mathcal{L}^a$. Statement 3) can be proved by standard constructions for ω-automata as in the aperiodic case [3, Theorem 6.4]. Concerning point 4) observe that $\mathcal{DML}^p = \mathcal{L}^p$ would imply $\mathcal{L}^p$ closed under complement which contradicts statement 3) of Corollary 5. To prove statement 5) note that a timed automaton in $\mathcal{DMT}^p$ can be determined from $\mathcal{A}_2$ which recognizes the complement of L_2. Hence the problem is reduced to testing whether $L_1 \cap L_2^c$ is empty which is decidable by Corollary 5. ■

Remark 4.1 The first statement of the previous proposition does not hold for "deterministic" timed automata with ϵ-transitions, i.e. the timed automata in $\mathcal{T}^p_\epsilon$ such that, for every pair of distinct moves $(q, \phi_1, \sigma_1, X_1, p_1)$, $(q, \phi_2, \sigma_2, X_2, p_2)$, the condition

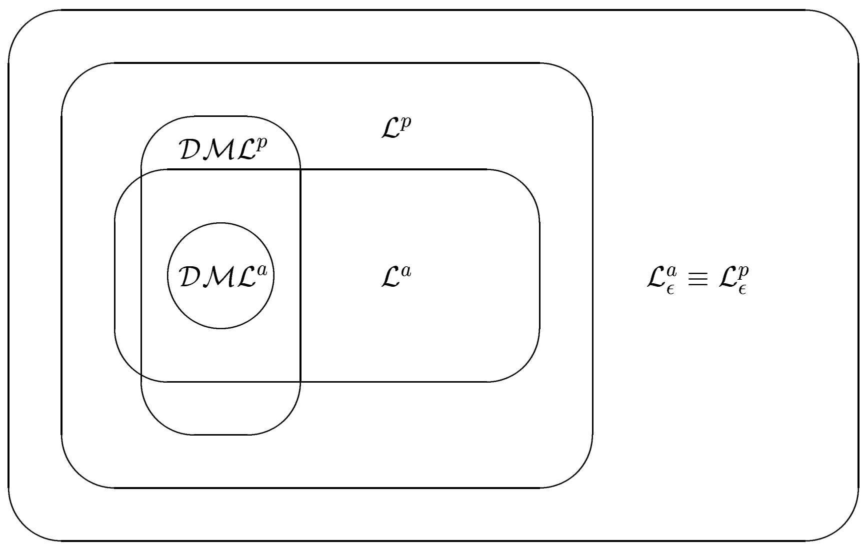
$$\sigma_1 = \sigma_2 \vee \sigma_1 = \epsilon \vee \sigma_2 = \epsilon$$

implies $\phi_1 \cap \phi_2 = \emptyset$ (definition first given in [2] for automata on aperiodic constraints). For instance, the timed automaton defined by the following picture



satisfies the previous property but any timed word $\{(a,t)\}$ with $t \in (1,2)$ admits two distinct runs. Hence, even ϵ-loops in timed automata induce a nondeterministic behaviour.

Concluding the comparison between automata with periodic and aperiodic constraints we summarize in the following picture the results of inclusion among different classes of timed languages presented in this section.



5 Removal of silent moves without clock-resets

In [11] the question is raised to determine under which conditions ϵ-transitions can be removed. Observe that for finite (untimed) automata, whether one or multi-tape, such moves can be eliminated while, as we have seen in the previous section, this is not true for timed automata. However, in the case of traditional clock constraints, it is proved in [11, 10] that ϵ-transitions with no clock reset can be suppressed: every restricted timed automaton in $\mathcal{T}^a_\epsilon$ is equivalent to a timed automaton in $\mathcal{T}^a$. Here, we show that the same result holds for timed automata with periodic clock constraints. This is due to the fact that the proof given in [11, 10] actually depends on the form of the constraints only through the following properties:

1. any constraint of a timed automaton in $\mathcal{T}_\epsilon^a$ may be assumed to be disjunction-free and diagonal-free;

2. ϵ-cycles can be eliminated from restricted timed automata in $\mathcal{T}_\epsilon^a$;

3. the family of traditional clock constraints is closed with respect to two special operations defined over subsets of $\mathbb{R}_+^n$, called forward and backward closure.

So, once statements 1, 2 and 3 above are proved for periodic constraints, the same proof presented in [11, 10] can be applied to the periodic case also, showing that silent moves can be removed from restricted timed automata in $\mathcal{T}_\epsilon^p$. Observe that the first two statements, for automata in $\mathcal{T}_\epsilon^p$, follow from Propositions 1 and 3, respectively. Hence, in order to extend the result we only have to prove statement 3 for periodic constraints.

5.1 Forward and backward closures

For every $A \subseteq \mathbb{R}^n$ and every $t \in \mathbb{R}$, let $A + t$ be the set given by

$$A + t = \{(x_1 + t, x_2 + t, \ldots, x_n + t) \in \mathbb{R}^n \mid (x_1, x_2, \ldots, x_n) \in A\}$$

and, for every $B \subseteq \mathbb{R}$, let $A + B$ denote the set $\bigcup_{t \in B} A + t$.

Then, given a subset $A \subseteq \mathbb{R}_+^n$, the *forward closure* $\overrightarrow{A}$ of A is defined by setting

$$\overrightarrow{A} = A + \mathbb{R}_+$$

Moreover, for every $I \subseteq \{1, \ldots, n\}$, the *backward closure* $\overleftarrow{A}^I$ of A relative to I is defined as the subset

$$\overleftarrow{A}^I = \pi_I^{-1}(A - \mathbb{R}_+)_I \tag{7}$$

where, for all $B \subseteq \mathbb{R}^n$, B_I is the set of elements $(x_1, \ldots, x_n) \in B \cap \mathbb{R}_+^n$ with $x_i = 0$ for all $i \in I$, and π_I is the projection of $\mathbb{R}_+^n$ onto the subspace determined by the equations $x_i = 0$ for all $i \in I$.

Clearly, $\overrightarrow{A}$ represents the family of clock assignments reachable from A after some delay; analogously, $\overleftarrow{A}^I$ is the set of all clock assignments from which one can reach A by applying a reset of the clocks indexed by I and waiting for some delay.

In [11, 10] it is proved that the family of traditional constraints is closed with respect to these operations. By a rather tiresome case study, a similar proof can be given for periodic constraints by using the canonical form described in Section 3.1. Such extension is presented in detail in Appendix B and allows to state the following

PROPOSITION **9** *Given a positive $n \in \mathbb{N}$, let $A \subseteq \mathbb{R}_+^n$ be defined by periodic constraints. Then, also $\overrightarrow{A}$ and $\overleftarrow{A}^I$ (for any $I \subseteq \{1, 2, \ldots, n\}$) can be defined by periodic constraints.*

By the previous discussion this proposition implies that silent moves in restricted periodic automata can be removed.

THEOREM **3** *Every restricted timed automaton in $\mathcal{T}_\epsilon^p$ admits an equivalent timed automaton in $\mathcal{T}^p$.*

Remark 5.1 Observe that, contrary to the case of aperiodic constraints [14], for automata in $\mathcal{T}_\epsilon^p$ the removal of ϵ-transition with clock resets not lying on a loop is not always possible. As an example consider the timed automaton described in the proof of Proposition 7: this contains a silent move outside any loop and accepts a timed language which cannot be recognized by any automaton in $\mathcal{T}^p$.

6 Count-down clocks

References to the future in the timed automata model exist in the literature. In [5], the authors aim at defining a class of timed languages on finite sequences that is closed with respect to the Boolean operations: the distance between the present time and the next occurrence of each input letter is "predicted". In [7] a different approach is proposed by allowing the input timed sequences to be scanned back and forth. Here, we introduce count-down clocks, i. e., clocks that decrease though running at the same speed as normal clocks. Also, each transition has two constraints that apply to the "count-up" and the "count-down" clocks separately. Here is a formal definition where $\overrightarrow{X}$ and $\overleftarrow{X}$ are two disjoint families of clocks, while $\Phi(\overrightarrow{X})$ and $\Phi(\overleftarrow{X})$ are the corresponding sets of periodic constraints.

DEFINITION 4 *A* timed (Büchi) automaton with count-down clocks *is a tuple* $\mathcal{A} = (\Sigma, Q, I, F, T, R, \overleftarrow{X}, \overrightarrow{X})$ *where* Σ, Q, I, F, R *are as in Definition 2,* $\overleftarrow{X}$ *is a finite set of real valued* count-up clocks, $\overrightarrow{X}$ *is a finite set of real valued* count-down clocks, $\overleftarrow{X} \cap \overrightarrow{X} = \emptyset$ *and* $T \subseteq Q \times \Phi(\overleftarrow{X}) \times \Phi(\overrightarrow{X}) \times (\Sigma \cup \{\epsilon\}) \times 2^{\overleftarrow{X}} \times 2^{\overrightarrow{X}} \times Q$ *is a finite set of* transitions.

The meaning of the transitions is a direct extension of the standard case. An *assignment* of the clocks is a function of $\nu : \overleftarrow{X} \cup \overrightarrow{X} \to \mathbb{R}_+$. A finite (resp. infinite) timed word $(\sigma_1, t_1)(\sigma_2, t_2) \ldots (\sigma_n, t_n)$ (resp. $(\sigma_1, t_1)(\sigma_2, t_2) \ldots (\sigma_i, t_i), \ldots$) is accepted by $\mathcal{A}$ if there exist

– a sequence of assignments ν_i, $i \geq 0$, where $\nu_0(x) = 0$ for all $x \in \overleftarrow{X}$ while $\nu_0(x)$ is an arbitrary value in $\mathbb{R}_+$ for every $x \in \overrightarrow{X}$,

– a sequence of states q_i, $i \geq 0$, with $q_0 \in I$, ending in $q_n \in F$ (resp. $q_i \in R$ for infinitely many i's),

– a sequence of transitions $(q_{i-1}, \phi_i, \sigma_i, \overleftarrow{X}_i, \overrightarrow{X}_i, q_i) \in T$, $i = 1, 2, \ldots$,

such that the following properties hold for every $i = 1, 2, \ldots$:

– for all $x \in \overleftarrow{X}$

$\nu_{i-1}(x) + t_i - t_{i-1}$ satisfies ϕ_i (with the initial condition $t_0 = 0$),

and $\nu_i(x) = \begin{cases} 0 & \text{if } x \in \overleftarrow{X}_i \\ \nu_{i-1}(x) + t_i - t_{i-1} & \text{otherwise;} \end{cases}$

– for all $x \in \overrightarrow{X}$

$\nu_{i-1}(x) = t_i - t_{i-1}$ if $x \in \overrightarrow{X}_i$,

$\nu_i(x) = \begin{cases} \text{arbitrary} & \text{if } x \in \overrightarrow{X_i} \\ \nu_{i-1}(x) - (t_i - t_{i-1}) & \text{otherwise,} \end{cases}$

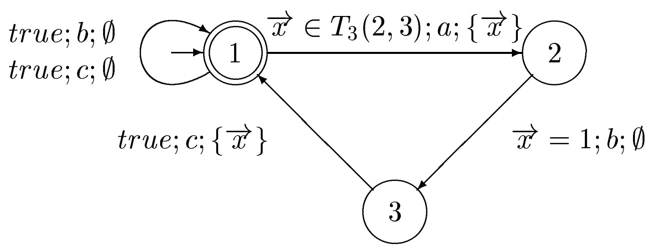
and $\nu_i(x)$ satisfies ϕ_i.

We denote by $\mathcal{T}_\epsilon^c$ the family of timed automata with count-down clocks and by $\mathcal{T}^c$ the subset of automata $\mathcal{A} \in \mathcal{T}_\epsilon^c$ without ϵ-transitions. Analogously, let $\mathcal{L}_\epsilon^c$ and $\mathcal{L}^c$ be the families of timed languages recognized by automata in $\mathcal{T}_\epsilon^c$ and $\mathcal{T}^c$, respectively.

EXAMPLE **6.1** The timed language $L \subseteq (\{a,b,c\} \times \mathbb{R}_+)^\infty$ of all strings $\{\sigma_i, t_i\}_{i\geq 1}$ such that, for every $i \geq 1$,

$$\sigma_i = a \;\Rightarrow\; \sigma_{i+1} = b, \sigma_{i+2} = c, t_{i+2} - t_i \in T_3(2,3), t_{i+2} = t_{i+1} + 1$$

is recognized by the following automaton that has just one count-down clock $\overrightarrow{x}$.



For instance, here is a run of the automaton over the timed string $(a, 10)(b, \frac{23}{2})(c, \frac{25}{2})$:

$$(1, \overrightarrow{x} = 10) \xrightarrow{a, t_1 = 10} (2, \overrightarrow{x} = \frac{5}{2}) \xrightarrow{b, t_2 = \frac{23}{2}} (3, \overrightarrow{x} = 1) \xrightarrow{c, t_3 = \frac{25}{2}} (1, \overrightarrow{x} = 3)$$

$\square$

Our main goal in this section is to prove that count-down clocks do not increase the expressive power of timed automata. We first present a technical lemma.

LEMMA **6.1** *Let $m \geq 3$ be an integer and, for each $1 \leq i \leq 2m$, let X_i be a finite, possibly empty, subset of real numbers such that for all $1 \leq i, j \leq 2m$ and all $x \in X_i, y \in X_j$, the following properties hold:*

1. *if* $1 \leq i, j \leq m$ *then*
$$i - j - 1 < x - y < i - j + 1$$
2. *if* $m < i$ *or* $m < j$ *then there exists an integer k such that*
$$i - j - 1 + km < x - y < i - j + 1 + km$$

Then, there exists a real number α such that $X_i \subseteq (\alpha + i - 1, \alpha + i)$ for all $1 \leq i \leq m$ and $X_i \subseteq T_m(\alpha + i - 1, \alpha + i)$ for all $m < i \leq 2m$.

Proof. Case 1: $\displaystyle\bigcup_{m < i \leq 2m} X_i = \emptyset$.

In this case we prove the statement by induction on the integer m. This is clear for $m = 1$. Now, let m be an arbitrary value and we want to prove the Lemma for $m+1$. Clearly, the statement holds if X_{m+1} is empty, so we assume it is not. Let α be the real satisfying the condition of the Lemma with the collection of X_i's for $1 \leq i \leq m$. Moreover, let α_1 and α_2 be defined by

$$\alpha_1 = \max\{x - i \mid i = 1, 2 \ldots, m, x \in X_i\},$$
$$\alpha_2 = \min\{x - i + 1 \mid i = 1, 2 \ldots, m, x \in X_i\}$$

Since $X_i \subseteq (\alpha + i - 1, \alpha + i)$ for all $1 \leq i \leq m$, it is easy to verify that $\alpha_1 < \alpha < \alpha_2$ and $0 < \alpha_2 - \alpha_1 \leq 1$. Furthermore, the following assertions hold for all $1 \leq i \leq m$:

1. $X_i \subseteq (\alpha_1 + i - 1, \alpha_1 + i]$ and there exist $1 \leq j_1 \leq m$ and $x_1 \in X_{j_1}$ such that $x_1 = \alpha_1 + j_1$,

2. $X_i \subseteq [\alpha_2 + i - 1, \alpha_2 + i)$ and there exist $1 \leq j_2 \leq n$ and $x_2 \in X_{j_2}$ such that $x_2 = \alpha_2 + j_2 - 1$.

Observe that for all values $\alpha_1 < \beta < \alpha_2$ we have $X_i \subseteq (\beta + i - 1, \beta + i)$ for all $0 < i \leq m$.

Now, for all $z \in X_{m+1}$, we have $m + 1 - j_1 - 1 < z - x_1 < m + 1 - j_1 + 1$ and $m + 1 - j_2 - 1 < z - x_2 < m + 1 - j_2 + 1$. Since $x_1 = \alpha_1 + j_1$ and $x_2 = \alpha_2 + j_2 - 1$, this yields respectively $m + \alpha_1 < z < m + 2 + \alpha_1$ and $m - 1 + \alpha_2 < z < m + 1 + \alpha_2$. As $\alpha_2 < \alpha_1 + 1$ holds, we obtain $n + \alpha_1 < z < m + 1 + \alpha_2$, i.e., $X_{m+1} \subseteq (n + \alpha_1, m + 1 + \alpha_2)$. Because $|x - y| < 1$ for all $1 \leq i \leq m+1$ and all $x, y \in X_i$, the subset X_{m+1} is contained in an interval of width 1. Thus, there exists $\alpha_1 < \beta < \alpha_2$ such that $X_{m+1} \subseteq (n+\beta, m+1+\beta)$ which by the above observation completes the verification in this case.

Case 2: $\bigcup_{m<i\leq 2m} X_i \neq \emptyset$.

Our idea is to assign an integer to each element of X_j, for all $m < j \leq 2m$, in such a way that two distinct elements of X_j are assigned the same integer if and only if they belong to the same connected component of the set $T_m(\alpha + j - 1, \alpha + j)$ that contains X_j. As shown in the following, this assignment allows to reduce the proof to Case 1.

First, assume there is some nonempty X_i, $1 \leq i \leq m$ and let $\bar{x} \in X_i$ be one of its elements. Then, for every $m < j \leq 2m$ and every $y \in X_j$, we assign to y the integer k given by the relation

$$j - i - 1 + km < y - \bar{x} < j - i + 1 + km \tag{8}$$

Such a k always exists by the hypothesis and we claim that its value does not depend on the choice of $\bar{x}$. Indeed, consider another element $x' \in X_{i'}$ for some $1 \leq i' \leq m$. Again, for some integer k', we have

$$j - i' - 1 + k'm < y - x' < j - i' + 1 + k'm \tag{9}$$

By point 1 of the hypothesis, we know that

$$i - i' - 1 < \bar{x} - x' < i - i' + 1 \tag{10}$$

while, from (8) and (9), we obtain

$$i - i' - 2 + (k' - k)m < \bar{x} - x' < i - i' + 2 + (k' - k)m \tag{11}$$

Now, combining (10) and (11) yields

$$\begin{gathered} i - i' - 1 < i - i' + 2 + (k' - k)m \\ i - i' - 2 + (k' - k)m < i - i' + 1 \end{gathered}$$

This implies $-3 < (k' - k)m < 3$ and thus $k = k'$ because of the hypothesis $m \geq 3$. Therefore, for every $y \in X_j$ with $m < j \leq 2m$, we have defined a unique integer k_y for which the inequality

$$j - i - 1 + k_y m < y - x < j - i + 1 + k_y m \tag{12}$$

holds for every $x \in X_i$ and every $1 \leq i \leq m$.

By a similar reasoning one can prove that, for every pair of elements $y \in X_j$ and $y' \in X_{j'}$ with $m < j, j' \leq 2m$, if $j - j' - 1 + \ell m < y - y' < j - j' + 1 + \ell m$ is satisfied then $\ell = k_y - k_{y'}$. Hence, we have

$$j - j' - 1 + (k_y - k_{y'})m < y - y' < j - j' + 1 + (k_y - k_{y'})m \tag{13}$$

for all such elements y, y'.

Thus, by inequalities (12) and (13), the sets $\hat{X}_i$, $1 \leq i \leq 2m$, such that $\hat{X}_i = X_i$ for all $1 \leq i \leq m$ and $\hat{X}_i = \{y - k_y m \mid y \in X_i\}$ for all $m < i \leq 2m$, satisfy hypothesis 1 of the lemma with m replaced by $2m$. As a consequence, the result follows from Case 1.

Assume now $\bigcup_{1 \leq i \leq m} X_i = \emptyset$. Let $\bar{x}$ be a arbitrarily fixed element in X_i for some $m < i \leq 2m$ and define the value $k_x = 0$. Moreover, for any other element $y \in X_j$, $m < j \leq 2m$, there exists an integer k such that

$$j - i - 1 + km < y - \bar{x} < j - i + 1 + km$$

and hence we can set $k_y = k$. Reasoning as above, we can show that for every pair of elements $y \in X_j$, $y' \in X_{j'}$, where $m < j, j' \leq 2m$, the inequality $j - j' - 1 + \ell m < y - y' < j - j' + 1 + \ell m$ implies $\ell = k_y - k_{y'}$. Therefore, we are again reduced to Case 1. ■

THEOREM 4 *We have $\mathcal{L}^c_\epsilon \equiv \mathcal{L}^p_\epsilon$ and $\mathcal{L}^c \equiv \mathcal{L}^p$.*

Proof. Let $\mathcal{A} = (\Sigma, Q, I, F, T, R, \overleftarrow{X}, \overrightarrow{X})$ be a timed automaton defined as above and let z be a count-down clock of $\mathcal{A}$. We describe a new automaton $\mathcal{A}'$, equivalent to $\mathcal{A}$, obtained by replacing z with new (count-up) clocks without modifying the other clocks and their constraints. Hence, the construction can be applied repeatedly to remove all count-down clocks of $\mathcal{A}$.

As in the case of ordinary clocks, we may suppose that there exists an integer m such that all z-constraints are defined as in Theorem 1. By a possible further refinement we can assume $m \geq 3$. Moreover, we omit the cases where the clock z satisfies an exact condition of the form $z = i$ or $z \in T_m[i]$; so, we consider only the z-constraints

$$\begin{aligned} &\phi_i(z) \equiv z \in (i, i+1), \\ &\phi_{i+m}(z) \equiv z \in T_m(i+m, i+m+1). \end{aligned}$$

Then, all constraints of $\mathcal{A}$ involving z are of the form $\phi \equiv \psi \wedge \phi_i(z)$ for some $0 \leq i < 2m$, where ψ is a condition on clocks other than z. We call *i-transition* a transition that bears a constraint of this form.

The new automaton $\mathcal{A}'$ has a set of states

$$Q' = \{(q, \alpha) \mid q \in Q, \alpha \subseteq \{0, 1, \cdots, 2m-1\}\}$$

In each state (q, α), α represents the indices i such that an i-transition has been crossed by the current run since the last reset of z.

Let us first explain the construction intuitively. Consider the portion of a run of $\mathcal{A}$ between two resets of z. This can be seen as a path over the transition diagram of $\mathcal{A}$

possibly crossing several i-transitions. It is simulated in the new automaton by a path of the same length that works in two phases: first it collects the information about the times of traversal of the i-transitions by using new (count-up) clocks, checking that the corresponding z-constraints are compatible with one another; then, the same constraints on these new clocks are contained in the last transition to simulate the condition $z = 0$ (i.e., the reset of z).

Formally, for each $i \in \{0, 1, \ldots, 2m-1\}$, let e_i and l_i be two new (ordinary) clocks; while $\mathcal{A}'$ simulates a path between two consecutive resets of z, these clocks have the following meaning:

– if $0 \leq i < m$, then e_i and l_i are reset respectively at the earliest and the latest move that simulates an i-transition. Hence, these clocks always satisfy the conditions $e_i \geq l_i$;

– if $m \leq i < 2m$, then e_i and l_i represent respectively the earliest and latest "modulo m" time of a traversal of an i-transition. More precisely, they are maintained as follows: they are both reset at the first move simulating an i-transition; then, to simulate any new i-transition at a given time one of the following conditions has to be satisfied

$$l_i \in T_m[0,1) \wedge e_i \in T_m[0,1) \tag{14}$$

$$l_i \in T_m(m-1,m) \wedge e_i \in T_m[0,1) \tag{15}$$

$$l_i \in T_m(m-1,m) \wedge e_i \in T_m(m-1,m) \tag{16}$$

The clocks l_i and e_i are reset respectively when conditions (14) and (16) occur. Note that, in this way, we have $e_i - l_i \in T_m[0,1)$ if $e_i \geq l_i$, while $l_i - e_i \in T_m(m-1,m)$, otherwise.

Now, let us define the transitions of $\mathcal{A}'$. These are of two types: those which simulate a move of $\mathcal{A}$ without reset of z and those that simulate such a reset. In the first case, the new constraint has to verify that the times elapsed from the previous i-transitions match with the z-constraint of the move to be simulated. Thus, the new constraint is given by hypotheses 1 and 2 of Lemma 6.1. The lemma itself guarantees the consistency of the process. On the contrary, the constraint of a move simulating a transition with reset of z simply verifies that the current values of all the clocks e_i and l_i satisfy their original z-constraints.

Formally, for every move $q \xrightarrow{\psi \wedge z \in (i,i+1);\sigma;X'} p$ in $\mathcal{A}$ (where ψ does not contain z), we define the following transitions of $\mathcal{A}'$:

1. if $z \notin X'$, the new moves are

$$(q,\alpha) \xrightarrow{\psi \wedge \phi;\sigma;Y} (p, \alpha \cup \{i\})$$

where $\alpha \subseteq \{i, i+1, \ldots, 2m-1\}$, such that either $i \notin \alpha$, $Y = X' \cup \{e_i, l_i\}$ and ϕ coincides with the constraint ϕ_1 given by

$$\phi_1 = \bigwedge_{j \in \alpha, j < m} e_j, l_j \in (j-i-1, j-i+1) \wedge \bigwedge_{j \in \alpha, j \geq m} e_j, l_j \in T_m(j-i-1, j-i+1),$$

or $i \in \alpha$, $Y = X' \cup \{l_i\}$ and $\phi = \phi_1 \wedge e_i \in [0,1)$. Observe that the atoms of ϕ coincides with the inequalities of the hypothesis of Lemma 6.1;

2. if $z \in X'$, the new moves are

$$(q, \alpha) \xrightarrow{\psi \wedge \phi_2; \sigma; Y_2} (p, \{i\})$$

where $\alpha \subseteq \{0, 1, \ldots, 2m-1\}$, such that

$$Y_2 = (X' \backslash \{z\}) \cup \bigcup_{j=0}^{2m-1} \{e_j, l_j\} \text{ and}$$

$$\phi_2 = \bigwedge_{j \in \alpha, j < m} e_j, l_j \in (j, j+1) \wedge \bigwedge_{j \in \alpha, j \geq m} e_j, l_j \in T_m(j, j+1)$$

Moreover, for every move $q \xrightarrow{\psi \wedge T_m(i,i+1); \sigma; X'} p$ in $\mathcal{A}$ (where ψ does not contain z and $m \leq i < 2m$), we define the following transitions of $\mathcal{A}'$:

1. if $z \notin X'$, the new moves are

$$(q, \alpha) \xrightarrow{\psi \wedge \phi; \sigma; Y} (p, \alpha \cup \{i\})$$

where $\alpha \subseteq \{m, m+1, \ldots, 2m-1\}$, such that either $i \notin \alpha$, $Y = X' \cup \{e_i, l_i\}$ and ϕ coincides with the constraint ϕ_3 given by

$$\phi_3 = \bigwedge_{j \in \alpha, j < i} e_j, l_j \in T_m(j-i+m-1, j-i+m+1) \wedge \bigwedge_{j \in \alpha, j > i} e_j, l_j \in T_m(j-i-1, j-i+1),$$

or $i \in \alpha$ and Y and ϕ are defined by one of the following three pairs of equations (corresponding to conditions (14), (15), (16), respectively):

$$Y = X' \cup \{l_i\}, \quad \phi = \phi_3 \wedge e_i, l_i \in T_m[0, 1)$$
$$Y = X', \quad \phi = \phi_3 \wedge e_i \in T_m[0, 1) \wedge l_i \in T_m(m-1, m)$$
$$Y = X' \cup \{e_i\}, \quad \phi = \phi_3 \wedge e_i, l_i \in T_m(m-1, m)$$

2. if $z \in X'$, the new moves are

$$(q, \alpha) \xrightarrow{\psi \wedge \phi_2; \sigma; Y_2} (p, \{i\})$$

where $\alpha \subseteq \{0, 1, \ldots, 2m-1\}$, while Y_2 and ϕ_2 are defined above.

At last, every transition $q \xrightarrow{\psi; \sigma; X'} p$ in $\mathcal{A}$ (where ψ does not involve z) defines the moves $(q, \alpha) \xrightarrow{\psi; \sigma; X'} (p, \alpha)$ with $\alpha \subseteq \{0, 1, \ldots, 2m-1\}$ if $z \notin X'$, and defines the moves $(q, \alpha) \xrightarrow{\psi \wedge \phi_2; \sigma; Y_2} (p, \emptyset)$ (with the same α), otherwise.

The initial states of $\mathcal{A}'$ are the states $(q, \emptyset)$ such that $q \in I$, while the final and repeated states are the pairs (q, α) such that $\alpha \subseteq \{0, 1, \ldots, 2m-1\}$, $q \in F$ and $q \in R$, respectively.

The general case, when $\mathcal{A}$ also contains exact atoms of the form $z = i$ and $z \in T_m[i]$, can be treated in a similar way. We can add a new component to the states of $\mathcal{A}'$ to keep track of the constraints of this type met since the latest reset of z. So, the states of the new automaton are now of the form (q, α, β) where q and α are as before, while β is again a subset of $\{0, 1, \ldots, 2m-1\}$. We also have to add a new (ordinary) clock $l_i^=$, for each $i \in \{0, 1, \ldots, 2m-1\}$, to give the time elapsed from the latest move simulating an "exact" i-transition. Analogously, the new transitions are not difficult to define, we only have to adapt the previous construction to the new state component and the new clocks. ∎

7 Special cases

In this section we study special cases concerning the reset of clocks on one side and the discrete time on the other.

7.1 Reset-free and pure reset automata

By *reset-free* automaton we mean a timed automaton where no transition performs a reset; at the contrary, in a *pure reset* automaton every transition resets all clocks. Observe that in both cases we may assume that the automaton has one single clock. We show that the inclusion (and therefore the equivalence) of languages recognized by timed automata is decidable under either hypothesis.

PROPOSITION **10** *The inclusion problem for timed languages defined by reset-free timed automata in $\mathcal{T}_\epsilon^p$ is decidable.*

Proof. Our purpose is to construct from a reset-free timed automaton $\mathcal{A} \in \mathcal{T}_\epsilon^p$ an ordinary Büchi automaton $\mathcal{A}'$ (on infinite strings) conveying the same information. This allows to reduce our problem to the inclusion problem of traditional ω-rational languages [21, 20].

First observe that, by Theorem 3, we may remove all silent moves from the automaton and hence we can suppose $\mathcal{A} \in \mathcal{T}^p$. Then, let x be the clock of $\mathcal{A}$ and, by Theorem 1, let us restrict to the case when all constraints in $\mathcal{A}$ are of the form: i) $x = i$, ii) $x \in (i, i+1)$, iii) $x \in T_m[i+m]$, or iv) $x \in T_m(i+m, i+m+1)$, where $0 \leq i < m$; let $\mathcal{C}$ be the set of all these constraints and note they are disjoint.

With every letter $\sigma \in \Sigma$ we associate the symbols σ_B, where $B \in \mathcal{C}$. The idea is to replace in any sequence $(\sigma^{\{1\}}, t_1)(\sigma^{\{2\}}, t_2) \ldots (\sigma^{\{\ell-1\}}, t_{\ell-1})\ (\sigma^{\{\ell\}}, t_\ell) \ldots$, the ℓ-th occurrence $(\sigma^{\{\ell\}}, t_\ell)$ by the symbol σ_B^ℓ such that $t_\ell \in B$.

The states of $\mathcal{A}'$ are the pairs (q, B), where q is a state of $\mathcal{A}$ and $B \in \mathcal{C}$. The initial states are the pairs $(q, x = 0)$ where q is initial in $\mathcal{A}$, while the final states are the pairs (q, B) with q final in $\mathcal{A}$. Moreover, (q, B) is a repeated state if q is repeated in $\mathcal{A}$ and B is a constraint in $\mathcal{C}$ of the form iii) or iv).

The transitions of $\mathcal{A}'$ are described by the following construction. Every move $q \xrightarrow{B,\sigma} p$ of $\mathcal{A}$ defines a given set of transitions in $\mathcal{A}'$:

1. if $B \equiv x = j$, then the transitions $(q, x = i) \xrightarrow{\sigma_B} (p, B)$, for all $i \leq j$, and the transitions $(q, x \in (i, i+1)) \xrightarrow{\sigma_B} (p, B)$, for all $i < j$, are in $\mathcal{A}'$;

2. if $B \equiv x \in (j, j+1)$, then all the transitions $(q, x = i) \xrightarrow{\sigma_B} (p, B)$ and $(q, x \in (i, i+1)) \xrightarrow{\sigma_B} (p, B)$, for $i \leq j$, are in $\mathcal{A}'$;

3. if either $B \equiv x \in T_m[j+m]$ or $B \equiv x \in T_m(j+m, j+m+1)$, then the transitions $(q, A) \xrightarrow{\sigma_B} (p, B)$ are in $\mathcal{A}'$ for all constraints $A \in \mathcal{C}$.

The Büchi automaton $\mathcal{A}'$ carries all the information about $\mathcal{A}$. To prove it, let Δ be the alphabet of all symbols σ_B such that $\sigma \in \Sigma$ and $B \in \mathcal{C}$. Define the function $f : (\Sigma \times \mathbb{R}_+)^\infty \longrightarrow \Delta^\infty$ by replacing in each timed sequence $\{(\sigma^\ell, t_\ell)\}_\ell$ every pair (σ^ℓ, t_ℓ) by the symbol σ_B^ℓ such that $t_\ell \in B$, and let $f(\epsilon) = \epsilon$.

Then, the proof proceeds through the following claims.

Claim 1. For every $z \in \Delta^\infty$, z is recognized by $\mathcal{A}'$ if and only if there exists $w \in (\Sigma \times \mathbb{R}_+)^\infty$ recognized by $\mathcal{A}$ such that $f(w) = z$.

In the case $z \in \Delta^*$ the statement can be proved by induction on the length of z. The "if" direction follows easily from the construction. For the other direction, observe that for every path in $\mathcal{A}'$ of the form

$$(q_0, x = 0) \xrightarrow{\sigma^1_{B_1}} (q_1, B_1) \xrightarrow{\sigma^2_{B_2}} (q_2, B_2) \cdots (q_{n-1}, B_{n-1}) \xrightarrow{\sigma^n_{B_n}} (q_n, B_n) \qquad (17)$$

we can find a sequence of times $t_1 \leq t_2 \leq \cdots \leq t_n$, such that $t_i \in B_i$ for every i, and a path in $\mathcal{A}$ of the form

$$q_0 \xrightarrow{B_1, \sigma^1} q_1 \xrightarrow{B_2, \sigma^2} q_2 \cdots q_{n-1} \xrightarrow{B_n, \sigma^n} q_n \qquad (18)$$

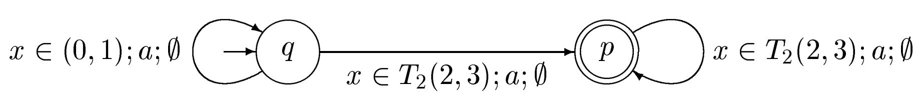
Then, if $z = \sigma^1_{B_1} \sigma^2_{B_2} \cdots \sigma^n_{B_n}$ is accepted by $\mathcal{A}'$ through the path (17), the sequence $w = (\sigma^1, t_1) \cdots (\sigma^n, t_n)$ is a timed word accepted by $\mathcal{A}$ through the path (18), and $f(w) = z$. A similar reasoning can be applied if z is an infinite string, recalling that every infinite $w \in (\Sigma \times \mathbb{R}_+)^\infty$ has a divergent time sequence.

Claim 2. For every $z \in \Delta^\infty$, if z is recognized by $\mathcal{A}'$ then $\mathcal{A}$ accepts every timed word $w \in (\Sigma \times \mathbb{R}_+)^\infty$ such that $f(w) = z$.

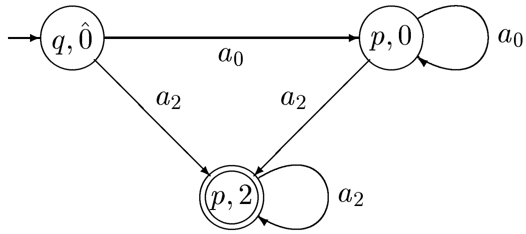
Indeed, by the previous claim, we know that there exists a timed word w accepted by $\mathcal{A}$ such that $f(w) = z$. Reasoning as before, one can show that any other timed string $w' \in (\Sigma \times \mathbb{R}_+)^\infty$, such that $f(w') = z$, is recognized in $\mathcal{A}$ through the same runs over w. As a consequence, also w' is recognized by the automaton.

Now, given two reset-free timed automata $\mathcal{A}$ and $\mathcal{B}$, let m be a common integer for which all constraints of $\mathcal{A}$ and $\mathcal{B}$ are of the form (3) and denote by $L(\mathcal{A})$ and $L(\mathcal{B})$ the corresponding languages. Then, construct as above the Büchi automata $\mathcal{A}'$ and $\mathcal{B}'$: applying the previous claims it turns out that $L(\mathcal{A})$ is included in $L(\mathcal{B})$ if and only if $L(\mathcal{A}')$ is included in $L(\mathcal{B}')$. ∎

Example **7.1** Let $\mathcal{A}$ be the automaton defined by the following picture:



It is easy to see that $\mathcal{A}$ recognizes the set of infinite sequences $\{(a, t_i)\}_{i \geq 1}$ such that $t_i \in T_2(0,1)$ for every $i \geq 1$ (recall that our infinite timed strings have divergent time sequence). The following picture describes the automaton $\mathcal{A}'$, obtained from $\mathcal{A}$ by the construction given in the previous proof (here, for brevity, the constraints $x = 0$, $x \in (0,1)$ and $x \in T_2(2,3)$ are represented by $\hat{0}$, 0 and 2, respectively).



Note that an isomorphic automaton is obtained from the timed automaton (equivalent to $\mathcal{A}$) consisting of a single (initial and repeated) state q together with the transition $q \xrightarrow{x \in T_2(0,1); a; \emptyset} q$. □

PROPOSITION **11** *The inclusion problem for timed languages defined by pure reset timed automata in $\mathcal{T}^p$ is decidable.*

Proof. We proceed in the same vein as in Proposition 10. In particular, the constraints of the automata are assumed to be of the same form as in the previous proof. Again, we associate with every letter $\sigma \in \Sigma$ the symbols σ_B for $B \in \mathcal{C}$. The idea is to replace, in any sequence $(\sigma^{\{1\}}, t_1)(\sigma^{\{2\}}, t_2) \ldots (\sigma^{\{\ell-1\}}, t_{\ell-1})(\sigma^{\{\ell\}}, t_\ell) \ldots$, the ℓ-th occurrence $(\sigma^{\{\ell\}}, t_\ell)$ by the symbol σ_B such that $t_\ell - t_{\ell-1} \in B$. Here, the construction of the new automaton is simpler: the states remain unchanged and the moves are of the form $q \xrightarrow{\sigma_B} p$ for every transition $q \xrightarrow{B, \sigma, \{x\}} p$ in the original automaton.

■

7.2 The discrete time

The notion of timed string and timed automaton given in Section 2 can be easily restricted to a discrete time domain. Under this assumption, we can obtain some further results similar to the properties presented in the previous section. They can be seen as extensions of some results in [8] and [17], the novelty being that we consider periodic constraints on one hand and ϵ-moves on the other.

Formally, a time sequence $\{t_i\}_i$ is *discrete* if $t_i \in \mathbb{N}$ for all indices i; we say that a timed string $\{(\sigma_i, t_i)\}_i \in (\Sigma \times \mathbb{R}_+)^\infty$ is *discrete* if $\{t_i\}_i$ is a discrete time sequence. The family of all discrete timed strings is denoted by $(\Sigma \times \mathbb{N})^\infty$. Whenever we assume $\mathbb{N}$ as time domain, the timed words accepted by a timed automaton are supposed to be discrete. Of course, in this case, we may assume integer the constraints of the automaton.

PROPOSITION **12** *Assuming $\mathbb{N}$ as time domain, each timed automaton with periodic clock contraints and possible ϵ-transitions is equivalent to a pure reset timed automaton with periodic clock contraints and no ϵ-transition.*

Proof. We will first show that the original automaton $\mathcal{A}$ can be transformed into an automaton $\mathcal{A}'$ with one single clock x that is reset after each transition. Number the clocks from 1 to n and assume further that the constraint associated with a transition is of the form $\bigwedge_{1 \le i \le n} \psi_i(x_i)$ where each condition ψ_i defines a rational subset $[\![\psi_i]\!]$ of $\mathbb{N}$. Consider the intersection $\sim$ of the right invariant equivalences associated with all the $[\![\psi_i]\!]$'s in all transitions. It exists and has finite index. Denote by $[m]$ the class of the integer m relative to $\sim$, by $[m]^{-1}[p]$ the set of integers $r > 0$ such that $m + r \sim p$. Then, the states of $\mathcal{A}'$ are the tuples of the form $(q, [a_1], [a_2], \ldots, [a_n])$, where q is a state in $\mathcal{A}$ and $a_1, a_2, \ldots, a_n$ are nonnegative integers.

Now, if $q \xrightarrow{\phi;\sigma;Y} p$ is a transition of $\mathcal{A}$, with $\sigma \in \Sigma \cup \{\epsilon\}$ and $\phi \equiv \bigwedge_{1\leq i\leq n} \psi_i(x_i)$, then for every $a_1, \ldots, a_n, b_1, \ldots, b_n \in \mathbb{N}$ the move

$$(q, [a_1], \ldots, [a_n]) \xrightarrow{\theta(x);\sigma;\{x\}} (p, [b_1], \ldots, [b_n])$$

is a transition of $\mathcal{A}'$ if there exist n equivalence classes $[c_1], \ldots, [c_n]$ such that:

1. $[c_i] \subseteq [\![\psi_i]\!]$, for all $i = 1, 2, \ldots, n$;
2. $\theta = \bigcap\limits_{1\leq i\leq n} [a_i]^{-1}[c_i]$;
3. $[b_i] = \begin{cases} [0] & \text{if } x_i \in Y \\ [c_i] & \text{otherwise,} \end{cases} \quad$ for all $i = 1, 2, \ldots, n$.

At this point, we may assume the automaton is pure reset and we prove that its ϵ-moves may be eliminated. An ϵ-path is a path taking ϵ-transitions only. The idea is to view an infinite path as an infinite sequence of ϵ-paths followed by a σ-transition where $\sigma \in \Sigma$. This means that we group together the ϵ-transitions preceding a given occurrence of a letter σ and that we add the duration of the ϵ-path to the duration of the σ-transition. More formally, let $q, p \in Q$ be two states. We distinguish the ϵ-paths according to whether or not they visit some repeated state. The set of all ϵ-paths taking q to p and not visiting any repeated state is rational over the alphabet of the transitions of the automaton. Now the set of durations of these paths is the image of this set by a rational substitution which assigns the clock constraint with each transition. Denote this last rational subset of $\mathbb{N}$ by $A_{q,p}$. Similarly, the subset $B_{q,p}$ of all durations of paths labelled by ϵ and taking q to p and visiting some repeated state is a rational subset of $\mathbb{N}$.

Now we modify the transitions of the automaton and we add a component to the state in order to remember whether or not the ϵ-path preceding a given occurrence $\sigma \in \Sigma$ has visited a repeated state. More specifically, the new set of states is $Q \times \{0, 1\}$ and the set of repeated states is $Q \times \{1\}$. We denote by $\phi^{\sigma}_{q,p}$ the time constraint associated with the σ-transition taking q to p. We define

$$(q, i) \xrightarrow{\psi;\sigma;\{x\}} (p, 0) \text{ where } [\![\psi]\!] = \bigcup_{r\in Q} (A_{q,r} + [\![\phi^{\sigma}_{r,p}]\!])$$

and

$$(q, i) \xrightarrow{\psi;\sigma;\{x\}} (p, 1) \text{ where } [\![\psi]\!] = \bigcup_{r\in Q} (B_{q,r} + [\![\phi^{\sigma}_{r,p}]\!])$$

(here the sum of two sets is to be understood as the set of all possible sums of their elements). Concerning the final states observe that it is decidable for a state $q \in Q$ whether there exists a final (possible empty) ϵ-path from q to a final state or an infinite ϵ-path starting from q and visiting a repeated state infinitely often. Hence the final states in the new automaton are the pairs (q, i) where q satisfies such a condition.

■

Since all transitions reset the clock, the same technique as in the previous pure-reset case applies and we can establish the following.

Corollary **13** *Assuming* $\mathbb{N}$ *as time domain, the inclusion (and therefore the equality) of two timed languages defined by timed automata with periodic clock contraints and possible* ϵ*-transitions is decidable.*

8 Appendix A

In this section we give the proof of Theorem 1. We start with an easy property.

Lemma **8.1** *Consider the subsets of the form* $T_\lambda(H)$*, where* $\lambda \in \mathbb{Q}_+$ *and* H *is a rational interval in* $\mathbb{R}_+$*. Then, the family of the finite unions of these sets forms a Boolean algebra.*

Proof. We first prove the closure under intersection. Since intersection distributes over union, it suffices to consider the expression $T_\lambda(H) \cap T_\mu(K)$. If either λ or μ equals 0 the property is a simple consequence of the definition. Otherwise, multiplying by the least common multiple of the denominators of the different constants, we may assume that H and K are integer intervals and $\lambda, \mu \in \mathbb{N}\backslash\{0\}$. Let ν be the least common multiple of λ and μ. Then, there exists a positive $p \in \mathbb{N}$ and four finite unions of integer intervals U, V, H', K' such that $U \cup V \subseteq [0,p)$, $H' \cup K' \subset [p, +\infty)$ and

$$T_\lambda(H) = U \cup T_\nu(H') \text{ and } T_\lambda(K) = V \cup T_\nu(K');$$

thus

$$T_\lambda(H) \cap T_\mu(K) = (U \cap V) \cup T_\nu(H' \cap K')$$

that is easily shown to be of the required form. Concerning the complement, it suffices to deal with the sets $T_\lambda(H)$ such that $H = [a,b]$ and $a, b, \lambda \in \mathbb{Q}_+$, $0 < a \leq b$ (the other cases can be treated similarly). Then, the complement satisfies the equality

$$\overline{T_\lambda(H)} = \begin{cases} [0,a) \cup T_1(b, b+1] & \text{if } \lambda = 0 \\ [0,a) & \text{if } 0 < \lambda \leq b - a \\ [0,a) \cup T_\lambda(b, a+\lambda) & \text{if } b - a < \lambda \end{cases}$$

which completes the proof. ■

Now, we are able to give the proof of the theorem. By Proposition 1 we may consider a timed automaton $\mathcal{A} \in \mathcal{T}_\epsilon^p$ with nondiagonal clock constraints. Then, each constraint ϕ in $\mathcal{A}$ can be represented in disjunctive normal form

$$\phi = \bigvee_{i \in I} \bigwedge_{j \in J_i} B_{ij} \tag{19}$$

where I and each J_i are finite sets of indices and every B_{ij} is a nondiagonal integer atom or the negation of such an atom. By Lemma 8.1 and via possible further decompositions, each B_{ij} is the finite disjunction of constraints of the form

$$x \in T_\lambda(H) \tag{20}$$

where $\lambda \in \mathbb{N}$, $H = [a]$ or $H = (a, a+1)$, with $a \in \mathbb{N}$.

Let p be a multiple of the nonzero λ's appearing in the B_{ij}'s of all constraints ϕ of $\mathcal{A}$. Then, each $T_\lambda(H)$ with $\lambda \neq 0$ is the finite union of sets of the form

$$\mu\lambda + T_p(H) = T_p(\mu\lambda + H) = T_p(K) \tag{21}$$

for some integer μ such that $0 \leq \mu < \frac{p}{\lambda}$. Observe that all $T_p(K)$'s obtained above have the same period p and for every $k \in K$ there exists $h \in H$ such that $k < h + p$.

Now, let $m = kp$ be a multiple of p greater than any constant appearing in all the previous intervals H and K. Then, applying the same transformation as in (21), each $T_p(K)$ can be reduced to the finite union of integer intervals and sets $T_m(U)$ such that $m \leq u < 2m$ for every $u \in U$. All constraints obtained in this way are of type (3); replacing them in each B_{ij} of equation (19) we have $\phi = \bigvee_{i=1}^{t} \phi_i$, where every ϕ_i is a conjunction of constraints of the form (3). Then, each transition (p, ϕ, σ, Y, q) of $\mathcal{A}$ can be replaced by the moves $(p, \phi_i, \sigma, Y, q)$, $i = 1, \ldots, t$. Applying the same substitution to all transitions of $\mathcal{A}$ we obtain an equivalent automaton $\mathcal{A}'$ as required. Observe that all decompositions performed in the construction are disjoint. Hence, if the original automaton is deterministic, so is the final one.

Observe that Lemma 3.1 is proved by applying the previous decompositions to the diagonal and nondiagonal atoms of any integer constraint represented in disjunctive normal form.

9 Appendix B

Our aim in this section is to prove Proposition 9. To this end, we assume X as ordered set of n clocks and ambiguously consider a periodic constraint either as a formula of type (1) over X, or as a subset of $\mathbb{R}_+^n$. The meaning will be clear from the context. Analogously, a clock assignment is an element $a \in \mathbb{R}_+^n$ and we denote by a_x the component of a corresponding to the clock $x \in X$.

We first fix some notations used in the following. For every $i, j, m \in \mathbb{N}$ we denote by $TR_m(i, j)$ the open triangles in $\mathbb{R}^2$ of vertices $(i+km, j)$, $(i+km+1, j)$, $(i+km+1, j+1)$ where $k \in \mathbb{N}$. Formally, we have

$$TR_m(i,j) = \{(x,y) \in \mathbb{R}^2 \mid y \in (j, j+1), \exists k \in \mathbb{N} : x < i + km + 1, x - y > i - j + km\}$$

We also consider a sort of semi-closed version of the previous set defined by

$$TR_m(i,j] = TR_m(i,j) \cup \{(x,y) \in \mathbb{R}^2 \mid y \in (j, j+1), x \in T_m[i+1]\}$$

It is easy to prove that both these sets are periodic constraints. For instance, any $(x, y) \in \mathbb{R}^2$ belongs to $TR_m(i, j)$ if and only if

$$x \in T_m(i, i+1) \wedge y \in (j, j+1) \wedge x - y \in T_m(i - j, i - j + 1)$$

This proves the following

LEMMA **9.1** *For every $i, j, m \in \mathbb{N}$, the sets $TR_m(i, j)$ and $TR_m(i, j]$ are integer periodic constraints.*

9.1 Forward closure

Now, we prove that the periodic constraints are closed with respect to the forward operation. Recall that, for any $A \subseteq \mathbb{R}_+^n$, a clock assignment $a \in \mathbb{R}_+^n$ belongs to $\overrightarrow{A}$ if and only if $a - t \in A$ for some $t \geq 0$ (here, $+$ and $-$ are defined as in Section 5.1).

We start by considering the forward closure of the conjunction of two constraints of the form (3).

Lemma **9.2** *Let A and B be two constraints of the form (3) over distinct clocks (x and y, respectively). Then, $\overrightarrow{A \wedge B}$ is the periodic constraint given by the following table.*

Case	A	B	$\overrightarrow{A \wedge B}$
1	$x = i$	$y = j$	$x \geq i \ \wedge \ x - y = i - j$
2	$x \in (i, i+1)$	$y = j$	$y \geq j \ \wedge \ x - y \in (i-j, i-j+1)$
3	$x \in T_m[i]$	$y = j$	$y \geq j \ \wedge \ x - y \in T_m[i-j]$
4	$x \in T_m(i, i+1)$	$y = j$	$y \geq j \ \wedge \ x - y \in T_m(i-j, i-j+1)$
5	$x \in (i, i+1)$	$y \in (j, j+1)$	$x > i \ \wedge \ y > j \ \wedge \ x - y \in (i-j-1, i-j+1)$
6	$x \in T_m[i]$	$y \in (j, j+1)$	$y > j \ \wedge \ x - y \in T_m(i-j-1, i-j) \ \wedge$ $\wedge \ \neg(x, y) \in TR_m(i-1, j)$
7	$x \in T_m(i, i+1)$	$y \in (j, j+1)$	$y > j \ \wedge \ x - y \in T_m(i-j-1, i-j+1) \ \wedge$ $\wedge \ \neg(x, y) \in TR_m(i-1, j]$
8	$x \in T_m[i]$	$y \in T_m[j]$	$\{x \geq i \ \wedge \ y - x \in T_m[j-i]\} \ \vee$ $\{y \geq j \ \wedge \ x - y \in T_m[i-j]\}$
9	$x \in T_m(i, i+1)$	$y \in T_m[j]$	$\{y \geq j \ \wedge \ x - y \in T_m(i-j, i-j+1)\} \ \vee$ $\vee \ \{x > i \ \wedge \ \neg(y, x) \in TR_m(j-1, i) \ \wedge$ $\wedge \ y - x \in T_m(j-i-1, j-i)\}$
10	$x \in T_m(i, i+1)$	$y \in T_m(j, j+1)$	$\{y > j \ \wedge \ x - y \in T_m(i-j-1, i-j+1) \ \wedge$ $\wedge \neg(x, y) \in TR_m(i-1, j]\} \ \vee$ $\vee \ \{x > i \ \wedge \ \neg(y, x) \in TR_m(j-1, i] \ \wedge$ $\wedge \ y - x \in T_m(j-i-1, j-i+1)\}$

The proof is obtained by considering the geometric picture of the constraints in each case.

Lemma **9.3** *Let $A \subseteq \mathbb{R}_+^n$ be a conjunction of constraints of the form (3) with $m \geq 3$. Then $\overrightarrow{A}$ is an integer periodic constraint.*

Proof. We assume that A is not empty and involves more than one clock, otherwise the property is obvious. Then, we have

$$A = \bigwedge_{x \in Y} A_x$$

where Y is a set of at least two clocks, each A_x is a constraint of the form (3) over the clock x. Further, all atoms A_x of type (iii) or (iv) in (3) have the same period $m \geq 3$. By Lemma 9.2 we know that $\overrightarrow{A_x \wedge A_y}$ is a periodic constraint for every $x \neq y$. As a consequence, so is

$$C = \bigwedge_{x \neq y} \overrightarrow{A_x \wedge A_y}$$

Then, in order to prove the lemma, it suffices to show $\overrightarrow{A} = C$.

First observe that $\overrightarrow{A} \subseteq C$ because, if $a \in \overrightarrow{\bigwedge_{x\in Y} A_x}$, then $a - t \in \bigwedge_{x\in Y} A_x$ for some $t \geq 0$; this implies $a - t \in A_x \cap A_y$ for all $x \neq y$, and hence $a \in C$.

On the other hand, let $a \in C$. With each $x \in Y$ associate the subset I_x of values $t \in \mathbb{R}$ such that $a \in A_x + t$. Each I_x has one of the following form according to the value of A_x:

$$\begin{array}{lll}
(a) & \{a_x - i_x\} & \text{if } A_x \equiv x = i_x \\
(b) & (a_x - i_x - 1, a_x - i_x) & \text{if } A_x \equiv x \in (i_x, i_x + 1) \\
(c) & \{a_x\} - T_m[j_x] & \text{if } A_x \equiv x \in T_m[j_x] \\
(d) & \{a_x\} - T_m(j_x, j_x + 1) & \text{if } A_x \equiv x \in T_m(j_x, j_x + 1)
\end{array}$$

Since $a \in C$, we have $\mathbb{R}_+ \cap I_x \cap I_y \neq \emptyset$ for all $x, y \in Y$. For a similar reason, in order to prove $a \in \overrightarrow{A}$, it suffices to show

$$\mathbb{R}_+ \cap \bigcap_{x\in Y} I_x \neq \emptyset \tag{22}$$

The inequality is true if there exists a subset I_x of type (a), for some $x \in Y$, because in this case I_x is reduced to a unique value that is necessarily common to all other subsets, so we rule out this case. Thus, define U as the set of all $x \in Y$ such that I_x is of type (b) and assume U nonempty. In this case, the nonempty set $I = \bigcap_{x\in U} I_x$ satisfies the equality

$$I = (a_u - i_u - 1, a_v - i_v)$$

where $u, v \in U$ are the clocks such that $a_u - i_u - 1 = \max\{a_x - i_x - 1 \mid x \in U\}$ and $a_v - i_v = \min\{a_x - i_x \mid x \in U\}$. Observe that $I = I_u \cap I_v$, while the interval $I_v \cup I_u = \bigcup_{x\in U} I_x$ has size smaller than 2.

Now, consider the subsets I_x of type (c) ((d), respectively). Since $m > 2$, these sets consist of distinct points (open intervals, resp.) whose mutual distances are greater than 2 (greater than or equal to 2, resp.). Thus, for every I_x of type (c) ((d), resp.), the two subsets $I_x \cap I_u \cap \mathbb{R}_+$ and $I_x \cap I_v \cap \mathbb{R}_+$ have a nonempty intersection, otherwise $I_x \cap \mathbb{R}_+$ would contain two distinct points (two intervals, resp.) at a distance smaller than 2 from each other. As a consequence, $I_x \cap I \cap \mathbb{R}_+ \neq \emptyset$ for every $x \in Y \backslash U$ and, since the size of I is smaller than or equal to 1, reasoning as above one can obtain

$$\bigcap_{x\in Y\backslash U} (I_x \cap I \cap R_+) \neq \emptyset$$

which proves inequality (22).

Finally, if all subsets are of type (c) or (d), we proceed as follows. Either there exist subsets of type (c) and then their minimal nonnegative element is common to all subsets whether of type (c) or (d). Else, there exist intervals of type (d) only. In this case, define $v \in Y$ as the clock such that $a_v - j_v = \min\{a_x - j_x \mid x \in Y\}$ and, for every $x \in Y$, set $I_{x,k} = (a_x - j_x - km - 1, a_x - j_x - km)$ for every $k \in \mathbb{N}$. It can be seen that, for every $x \in Y$, there exists a unique $k_x \in \mathbb{N}$ such that $I_{x,k_x} \cap I_{v,0} \cap \mathbb{R}_+ \neq \emptyset$ (in particular $k_v = 0$). As a consequence, $\mathbb{R}_+ \cap \bigcap_{x\in Y} I_{x,k_x}$ is nonempty and this proves inequality (22).

∎

Proposition 14 *If $\phi \subseteq \mathbb{R}_+^n$ is an integer periodic constraint, then also $\overrightarrow{\phi}$ is a constraint of the same type.*

Proof. At the cost of a possible refinement, we may assume ϕ defined as in Lemma 3.1 with $m \geq 3$. Since $\overrightarrow{A \vee B} = \overrightarrow{A} \vee \overrightarrow{B}$ for all subsets $A, B \subseteq \mathbb{R}_+^n$, we can also restrict ϕ to the form

$$\phi = \bigwedge_{i \in I} A_i \wedge \bigwedge_{j \in J} B_j$$

where I and J are finite sets of indices, each A_i is a constraint of the form (3) and each B_j is of the form (4). Observe that, for every diagonal constraint B, $a \in \overrightarrow{B}$ if and only if $a - t \in B$ for every $t \geq 0$. This implies

$$\overrightarrow{\phi} = \overrightarrow{\bigwedge_{i \in I} A_i} \wedge \bigwedge_{j \in J} B_j$$

and hence the proposition follows from Lemma 9.3. ■

9.2 Backward closure

Now, it is the turn of the backward closure. We use the same notation as in the previous subsection. Moreover, for any clock assignment a and every $I \subseteq X$, let a_I denote the assignment b such that $b_x = 0$ if $x \in I$, while $b_x = a_x$ if $x \notin I$. Then, for every subset $A \subseteq \mathbb{R}_+^n$, $a \in \overleftarrow{A}^I$ if and only if $a_I + t \in A$ for some $t \geq 0$.

We proceed as in the previous subsection. First, by a simple but boring analysis, one can prove that for every $I \subseteq X$, if A and B are constraints of the form (3), then $\overleftarrow{A \wedge B}^I$ is a periodic constraint. We omit the proof of this fact and only show a typical case in the following table.

I	A	B	$\overleftarrow{A \wedge B}^I$
$x \in I, y \notin I$	$x \in T_m(i, i+1)$	$y \in T_m(j, j+1)$	$y \in T_m(j-i-1, j-i+1) \wedge$ $\wedge\ x \geq 0\ \wedge\ y \geq 0$

Then, let us consider a constraint of the form

$$A = \bigwedge_{x \in Y} A_x$$

where $Y \subseteq X$ has cardinality grater than 1, each A_x is a constraint of the form (3) over the clock x, and every A_x of type (iii) or (iv) have the same period $m \geq 3$. We can define the constraint

$$C = \bigwedge_{x \neq y} \overleftarrow{A_x \wedge A_y}^I$$

and prove that $\overleftarrow{A}^I = C$. By the observation above this proves that $\overleftarrow{A}^I$ is a periodic constraint. Indeed, $\overleftarrow{A}^I \subseteq C$ because $a \in \overleftarrow{A}^I$ implies $a_I + t \in A_x$ for every $x \in Y$ and some $t \geq 0$, and hence $a \in \overleftarrow{A}_x^I$ for all $x \in Y$, proving $a \in C$. On the other hand, given $a \in C$, for every $x \in Y$ let $I_x \subseteq \mathbb{R}$ be defined by

$$I_x = \begin{cases} A_x & \text{if } x \in I \\ A_x - a_x & \text{otherwise} \end{cases}$$

Observe that $t \in I_x$ for some $t \geq 0$ if and only if $a \in \overleftarrow{A_x}^I$. Analogously, since $a \in C$, we have $\mathbb{R}_+ \cap I_x \cap I_y \neq \emptyset$ for every $x \neq y$. Now, reasoning as in the proof of Lemma 9.3, one can prove that $\mathbb{R}_+ \cap \bigcap_{x \in Y} I_x \neq \emptyset$ (note that our sets I_x and the sets I_x considered in Lemma 9.3 have a similar form). Hence $a \in \overleftarrow{\bigwedge_{x \in Y} A_x}^I = \overleftarrow{A}^I$ and we have proved the following

LEMMA **9.4** *Let $A \subseteq \mathbb{R}_+^n$ be a conjunction of constraints of the form (3) with $m \geq 3$ and let $I \subseteq \{1, 2, \ldots, n\}$. Then $\overleftarrow{A}^I$ is an integer periodic constraint.*

Now, we are able to prove the main result of this subsection.

PROPOSITION **15** *For every integer periodic constraint $\phi \subseteq \mathbb{R}_+^n$ and every $I \subseteq \{1, 2, \ldots, n\}$ also $\overleftarrow{\phi}^I$ is an integer periodic constraint.*

Proof. We may assume ϕ defined as in Lemma 3.1 with $m \geq 3$. Since $\overleftarrow{A \vee B}^I = \overleftarrow{A}^I \vee \overleftarrow{B}^I$ for all subsets $A, B \subseteq \mathbb{R}_+^n$ and every $I \subseteq \{1, 2, \ldots, n\}$, we can further restrict ϕ to the form

$$\phi = \bigwedge_{i \in \alpha} A_i \wedge \bigwedge_{j \in \beta} B_j$$

where α and β are finite sets of indices, each A_i is a constraint of type (3) and each B_j is a constraint of type (4).

Since the B_j's are diagonal constraints, from the definition of backward closure, one can prove that for any $j \in \beta$

$$a \in \overleftarrow{B_j}^I \Longleftrightarrow a_I \in B_j \Longleftrightarrow a \in \overline{B_j}$$

where $\overline{B_j}$ is obtained from B_j by replacing each clock $x \in I$ by 0. Reasoning in the same way, all B_j can be moved away from $\overleftarrow{\phi}^I$ proving that

$$\overleftarrow{\phi}^I = \overleftarrow{\bigwedge_{i \in \alpha} A_i}^I \wedge \bigwedge_{j \in \beta} \overline{B_j}$$

and hence the proposition follows from Lemma 9.4. ■

References

[1] R. Alur, C. Courcoubetis, and D.L. Dill. Model-checking in Dense Real-time. *Inform. and Comput.*, 104:2–34, 1993.

[2] R. Alur and D.L. Dill. Automata for modelling real-time systems. In *Proceedings 17th ICALP*, number 443 in Lecture Notes in Computer Science, pages 322–335. Springer Verlag, 1990.

[3] R. Alur and D.L. Dill. A theory of timed automata. *Theoret. Comput. Sci.*, 126:183–235, 1994.

[4] R. Alur, T. Feder, and T.A. Henzinger. The Benefits of Relaxing Punctuality. *J. Assoc. Comput. Mach.*, 43:116–146, 1996.

[5] R. Alur, L. Fix, and T.A. Henzinger. A determinizable class of timed automata. In *Proceedings of CAV'94*, number 818 in Lecture Notes in Computer Science, pages 1–13. Springer Verlag, 1994.

[6] R. Alur and T.A. Henzinger. Logics and models of Real time: a survey. In *Proceedings of the REX Workshop on Real Time: Theory and Practice*, number 600 in Lecture Notes in Computer Science, pages 74–106. Springer Verlag, 1991.

[7] R. Alur and T.A. Henzinger. Back to the future: towards a theory of timed regular languages. In *Proceedings of FOCS'92*, pages 177–186, 1992.

[8] R. Alur and T.A. Henzinger. Real-time logics: Complexity and Expressiveness. *Inform. and Comput.*, 104:35–77, 1993.

[9] R. Alur and T.A. Henzinger. A really temporal logic. *J. Assoc. Comput. Mach.*, 41:181–204, 1994.

[10] B. Bérard, V. Diekert, P. Gastin, and A. Petit. Characterization of the expressive power of silent transitions in timed automata. *Fundamenta Informaticae*, 36:145–182, 1998.

[11] B. Bérard, P. Gastin, and A. Petit. On the power of non observable actions in timed automata. In *Proceedings of STACS '96*, number 1046 in Lecture Notes in Computer Science, pages 247–258. Springer Verlag, 1996; also appeared in extended version as Technical Report LITP97/23, Université Paris 7, 1997.

[12] B. Bérard and C. Picaronny. Accepting Zeno words without stopping time. In *Proceedings of MFCS'97*, number 1295 in Lecture Notes in Computer Science, pages 149–158. Springer Verlag, 1997.

[13] F. Demichelis and W. Zielonka. Control timed automata. In *Proceedings of CONCUR '98*, number 1466 in Lecture Notes in Computer Science, pages 455–469. Springer Verlag, 1998.

[14] V. Diekert, P. Gastin, and A. Petit. Removing ϵ-transitions in timed automata. In *Proceedings of STACS'97*, number 1200 in Lecture Notes in Computer Science, pages 583–594. Springer Verlag, 1997.

[15] V. Gupta, T. A. Henzinger, and R. Jagadeesan. Robust timed automata. In *Proceedings of HART'97*, number 1201 in Lecture Notes in Computer Science, pages 331–345. Springer Verlag, 1997.

[16] T.A. Henzinger. The theory of hybrid automata. In *Proceedings of LICS'96*, pages 278 –292, New Brunswick, New Jersey, 1996. Invited tutorial.

[17] T.A. Henzinger, P.W. Kopke, and H Wong-Toi. The expressive power of clocks. In *Proceedings of ICALP'95*, number 944 in Lecture Notes in Computer Science, pages 335–346. Springer Verlag, 1995.

[18] P. Hermann. Automates temporisés et reconnaissabilité. *Inform. Process. Lett.*, 65:313–318, 1998.

[19] Z. Manna and A. Pnueli. Models for reactivity. *Acta Informatica*, 30(2):609–678, 1993.

[20] D. Perrin and J.-E. Pin. Mots infinis. Technical report, LITP 97/04, Institut Blaise Pascal, Université Paris 7, 1997.

[21] W. Thomas. Automata on infinite objects. In J. Van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, pages 133–191. Elsevier, Amsterdam, 1990.

[22] T. Wilke. Specifying timed state sequences in powerful decidable logics and timed automata. In H. Langmaack, W.-P. de Roever, and J. Vytopil, editors, *Formal Techniques in Real-Time and Fault-Tolerant Systems*, volume 863 of *Lecture Notes in Computer Science*. Springer Verlag, 1994.