# A Semi-decision Procedure for Testing Language Inclusion of Nondeterministic Timed Automata

*Arnaldo V. Moura and Guilherme A. Pinto*
{arnaldo,guialbu}@dcc.unicamp.br

**Relatório Técnico IC–00-02**

Janeiro de 2000

# A Semi-decision Procedure for Testing Language Inclusion of Nondeterministic Timed Automata

Arnaldo V. Moura and Guilherme A. Pinto
{arnaldo,guialbu}@dcc.unicamp.br

### Abstract

We give a new semi-decision procedure for testing language inclusion of nondeterministic timed automata (NTA). We show that the language generated by a *progressive* timed automaton can be tested for inclusion against the language generated by *any* NTA. In practice, many timed automata models of actual physical systems are progressive, so that the full expressiveness of NTA can be used to specify real-time properties. These include models of asynchronous digital circuits. The semi-decision procedure is also a reduction of the language inclusion problem for NTA to the language inclusion problem for nondeterministic effective infinite-state $\omega$-automata.

## 1 Introduction

Timed automata (TA) were proposed in [1] as a formalism for the verification of real-time systems. The formalism has been extensively studied and applied to practical problems. In the general verification problem, the system and the specification (the desired property) are modeled as TA, so that the problem reduces to testing language inclusion, which is undecidable for nondeterministic timed automata (NTA) [1]. One solution, frequently proposed in the literature, is to use, for the specification, a less expressive formalism, in such a way that the problem becomes decidable. Two such formalisms are deterministic TA [1] and event-clock TA [2]. On the other hand, the notion of nondeterminism facilitates the specification of properties and gives rise to, potentially, smaller models. For these reasons, the investigation of more powerful decision procedures for NTA is a problem of considerable interest.

In this paper, we give a new semi-decision procedure for testing language inclusion of arbitrary NTA. The procedure generalizes the region graph [1] used to solve the emptiness problem. It consists of a subset construction over a parallel composition of the two automata. The composition is guided by the system model automaton, and the two automata synchronize through a set of common *generic* clocks. The undecidability appears in the fact that the system and the specification may synchronize in such a way that an unbounded number of generic clocks is needed. However, we can show that the language generated by a *progressive* TA can be tested for inclusion against the language generated by *any* NTA. In practice, many TA models of actual physical systems are progressive. These include models of asynchronous digital circuits [3, 7]. In addition, the semi-decision procedure is also a reduction of the language inclusion problem for NTA to the language inclusion problem for nondeterministic effective infinite-state $\omega$-automata [10].

The paper is organized as follows. In Section 2 we review the formalism of TA. Section 3 presents the generalization of the region graph, and gives an example for which the semi-decision procedure does not terminate. In Section 4 we consider the progress condition under which the procedure will always terminate. Section 5 discusses the problem reduction to infinite-state $\omega$-automata, and Section 6 concludes with some final remarks.

## 2 Timed Automata

Informally, a timed automaton is a finite-state $\omega$-automaton (see Section 4) together with a finite set of clock variables whose values increase with the passage of time. Every transition of the automaton has a constraint on the values of the clocks and they can be taken only if the clocks satisfy the constraint. In addition, a transition may reset some of the clocks. TA accept timed words instead of $\omega$-words. A *timed word* $\rho$, over a finite alphabet of symbols $\Sigma$, is a pair $(\sigma, \tau)$ where: $\sigma = \sigma_1\sigma_2\cdots$ is a sequence of symbols $\sigma_i \in \Sigma$ (an $\omega$-word over $\Sigma$); and $\tau = \tau_1\tau_2\cdots$ is an strictly increasing sequence of time values $\tau_i \in \mathbb{R}$ (the set of non-negative real numbers), $\tau_i > 0$, satisfying the *progress* property: for every $t \in \mathbb{R}$, there is some $i \geq 1$ such that $\tau_i > t$. In a timed word $(\sigma, \tau)$, the time value $\tau_i$ is interpreted as the time when event $\sigma_i$ occurs. Given a finite set $X$ of clock variables, a *clock constraint* $\delta$ over $X$ is defined inductively by $\delta := x \leq c \,|\, c \leq x \,|\, \neg\delta \,|\, \delta_1 \wedge \delta_2$, where $x \in X$ and $c \in \mathbb{Q}$ (the set of non-negative rational numbers). The set of all clock constraints over $X$ is denoted by $\Phi(X)$.

A *timed Büchi automaton* $\mathcal{A}$ is a tuple $\langle \Sigma, Q, Q_0, X, T, F\rangle$, where

- $\Sigma$ is a finite alphabet of symbols;
- $Q$ is a finite set of locations;
- $Q_0 \subseteq Q$ is a set of start locations;
- $X$ is a finite set of clocks;
- $T \subseteq Q \times Q \times \Sigma \times 2^X \times \Phi(X)$ is a set of transitions. For a transition $\langle q, q', a, \lambda, \delta\rangle$ from location $q$ to location $q'$, on symbol $a$, $\delta$ gives the constraint to be satisfied and $\lambda$ gives the set of clocks to be reset;
- $F \subseteq Q$ is a set of accepting locations.

The language accepted by $\mathcal{A}$ is obtained by defining runs of $\mathcal{A}$ over timed words. For this, let a *clock interpretation for* $X$ be a function from $X$ to $\mathbb{R}$, that is, a particular reading of the clocks in $X$. A *generalized location* of $\mathcal{A}$ has the form $\langle q, \nu\rangle$, where $q \in Q$ and $\nu$ is a clock interpretation for $X$. For $t \in \mathbb{R}$, we write $\nu + t$ for the clock interpretation which maps every clock $x$ to $\nu(x) + t$. A clock interpretation $\nu$ for $X$ *satisfies* a clock constraint $\delta$ over $X$ iff $\delta$ evaluates to true when each clock $x$ is replaced by $\nu(x)$.

A *run* $r = (\overline{q}, \overline{\nu})$, of a TA $\mathcal{A}$ over a timed word $\rho = (\sigma, \tau)$ is an infinite sequence of generalized locations of the form $r : \langle q_0, \nu_0\rangle \rightarrow \langle q_1, \nu_1\rangle \rightarrow \langle q_2, \nu_2\rangle \rightarrow \;\cdots$, satisfying:

- *Initiation*: $q_0 \in Q_0$, and $\nu_0(x) = 0$ for all $x \in X$;
- *Consecution*: for all $i \geq 1$, there exists $\langle q_{i-1}, q_i, \sigma_i, \lambda_i, \delta_i \rangle \in T$ such that $(\nu_{i-1}+\tau_i-\tau_{i-1})$ satisfies $\delta_i$, and $\nu_i(x) = 0$ if $x \in \lambda_i$ and $\nu_i(x) = \nu_{i-1} + \tau_i - \tau_{i-1}$ otherwise ($\tau_0 = 0$, by definition).

Given a run $r = (\overline{q}, \overline{\nu})$ over a timed word $\rho = (\sigma, \tau)$, let $\mathit{inf}(r)$ be the set of locations such that $q \in \mathit{inf}(r)$ iff $q = q_i$ for infinitely many $i \geq 1$. The run $r$ over $\rho$ is called an *accepting* run iff $\mathit{inf}(r) \cap F \neq \emptyset$. Finally, the language accepted by $\mathcal{A}$ is $L(\mathcal{A}) = \{(\sigma, \tau) \mid \mathcal{A} \text{ has an accepting run over } (\sigma, \tau)\}$.

One natural way to define the verification problem is to model both the system and the specification (the desired property) as TA. Throughout the paper, $\mathcal{A}_1$ and $\mathcal{A}_2$ always denote the TA giving the system and the specification, respectively. The system satisfies the specification iff $L(\mathcal{A}_1) \subseteq L(\mathcal{A}_2)$. For deterministic[1] $\mathcal{A}_2$, the language inclusion problem reduces to testing emptiness of $L(\mathcal{A}_1) \cap \overline{L(\mathcal{A}_2)}$, which is decidable [1].

The emptiness problem for a TA $\mathcal{A}$ reduces to searching for a special cycle in a so called region graph, which is constructed from an equivalence relation on the set of generalized locations of $\mathcal{A}$ [1]. In the next section, we define a generalization of this region graph, which can be used, in many cases, to decide the language inclusion problem for NTA.

## 3 The Subset Construction Region Graph

Let $\mathcal{A}_1 = \langle \Sigma, Q_1, Q_{01}, X_1, T_1, F_1 \rangle$ and $\mathcal{A}_2 = \langle \Sigma, Q_2, Q_{02}, X_2, T_2, F_2 \rangle$. As in [1], we assume, without loss of generality, that all the constants in all the clock constraints of $\mathcal{A}_1$ and $\mathcal{A}_2$ are integers. We also assume that $\mathcal{A}_1$ and $\mathcal{A}_2$ are disjoint, except for the alphabet $\Sigma$. Since we cannot complement $\mathcal{A}_2$ in general [1], in order to cope with the nondeterminism we use the standard idea of a subset construction, applied on a parallel composition of the generalized locations of $\mathcal{A}_1$ and $\mathcal{A}_2$. We will not formally define the parallel composition or the subset construction. These concepts will be implicitly used in the definition of a graph $G$ over which the semi-decision procedure is obtained. The composition is guided by $\mathcal{A}_1$, that is, we take care of only the timed words which have some run of $\mathcal{A}_1$ over it. This is because $L(\mathcal{A}_1) \subseteq L(\mathcal{A}_2)$ iff every timed word that has an accepting run of $\mathcal{A}_1$ over it, also has an accepting run of $\mathcal{A}_2$ over it.

Let $A_1$ and $A_2$ denote, respectively, the set of all generalized locations of $\mathcal{A}_1$ and $\mathcal{A}_2$. The basic mathematical object used, from now on, is what we call a *composite pair* for $A_1$ and $A_2$, which has the form $\langle p, s \rangle$, where $p$ is a finite subset of $A_1$, and $s$ is a finite subset of $A_2$. We denote by $\mathcal{P}$ the set of all composite pairs for $A_1$ and $A_2$.

[1] A timed automaton is said to be deterministic iff (1) $|Q_0| = 1$, and (2) given any two transitions $\langle q_1, q_1', a_1, \lambda_1, \delta_1 \rangle$ and $\langle q_2, q_2', a_2, \lambda_2, \delta_2 \rangle$ in $T$, if $q_1 = q_2$ and $a_1 = a_2$, then $\delta_1 \wedge \delta_2$ is unsatisfiable. The interesting property of every deterministic timed automaton is that they have at most one run over every timed word.

### 3.1 Generic Clocks and the Equivalence Relation

The set $\mathcal{P}$ is uncountable. As in [1], we define an equivalence relation $\sim$ over $\mathcal{P}$ from which we obtain the graph $G$. It will turn out that the number of equivalence classes in $\sim$ is countable, but it is *not* finite. However, for many interesting instances of the language inclusion problem, $G$ will be finite, a fact that will guarantee termination of the procedure. In order to define the equivalence relation, we need to introduce the idea of a generic clock. We start with the following discussion.

Consider a finite timed word $\varrho = (\sigma_1, \tau_1) \to (\sigma_2, \tau_2) \to \cdots \to (\sigma_i, \tau_i)$. Let $\langle p_\varrho, s_\varrho \rangle$ be a composite pair, where $p_\varrho = \{\langle q, \nu \rangle \mid$ there is a finite run $\langle q_0, \nu_0 \rangle \to \langle q_1, \nu_1 \rangle \to \langle q_2, \nu_2 \rangle \to \cdots \to \langle q, \nu \rangle$ of $\mathcal{A}_1$ over $\varrho\}$; and let $s_\varrho$ be defined in the same way for $\mathcal{A}_2$. Also, let $D_{\langle p_\varrho, s_\varrho \rangle} = \{t \in \mathbb{R} \mid t$ is in the range of some clock interpretation in $p_\varrho$ or in $s_\varrho\}$. The composite pair $\langle p_\varrho, s_\varrho \rangle$ records enough information to determine the future behavior of $\mathcal{A}_1$ and $\mathcal{A}_2$ over any timed word having $\varrho$ as a prefix. Now consider another finite timed word $\varrho' = (\sigma'_1, \tau'_1) \to (\sigma'_2, \tau'_2) \to \cdots \to (\sigma'_{i+1}, \tau'_{i+1})$, such that $\varrho'$ has $\varrho$ as a prefix. Then, $|p_{\varrho'}|$ can be as high as $k_1$ times $|p_\varrho|$, where $k_1$ is the degree of nondeterminism[2] of $\mathcal{A}_1$. The same is true for $|s_{\varrho'}|$, $k_2$ and $|s_\varrho|$. It is interesting to note, however, that $|D_{\langle p_{\varrho'}, s_{\varrho'} \rangle}|$ is, at most, $|D_{\langle p_\varrho, s_\varrho \rangle}| + 1$.

Let $\alpha$ denote the greatest constant appearing in the clock constraints of $\mathcal{A}_1$ and $\mathcal{A}_2$. A value $t \in \mathbb{R}$ is called *relevant* if $t \leq \alpha$, and *irrelevant* otherwise. The above discussion motivates the definition of a generic clock. Informally, given a composite pair $\langle p, s \rangle$, we interpret each relevant value in $D_{\langle p,s \rangle}$ as being held by a generic clock; and all the irrelevant values in $D_{\langle p,s \rangle}$ as being held by a single generic clock. The traditional equivalence relation over the set of clock interpretations [1] is, instead, applied over the set of generic clock interpretations. Let us formalize these notions.

**Generic Clocks.** Let $\uparrow$ be a special symbol representing any value in the interval $(\alpha, \infty)$. By definition, $\uparrow > \alpha$. Given a composite pair $\langle p, s \rangle$, we define the set $R_{\langle p,s \rangle} \subset [0, \alpha] \cup \{\uparrow\}$ as follows: let $R'_{\langle p,s \rangle} = \{d \in \mathbb{R} \mid d \in D_{\langle p,s \rangle}$ and $d$ is relevant$\}$. If there is an irrelevant value in $D_{\langle p,s \rangle}$, then $R_{\langle p,s \rangle} = R'_{\langle p,s \rangle} \cup \{\uparrow\}$, otherwise $R_{\langle p,s \rangle} = R'_{\langle p,s \rangle}$. We create a set of generic clock variables, $C_{\langle p,s \rangle} = \{c_1, c_2, \ldots, c_{|R_{\langle p,s \rangle}|}\}$ for $\langle p, s \rangle$. The *generic clock interpretation* $\eta_{\langle p,s \rangle}$ is defined as the unique bijective function $\eta_{\langle p,s \rangle} : C_{\langle p,s \rangle} \to R_{\langle p,s \rangle}$ satisfying $\eta_{\langle p,s \rangle}(c_1) < \eta_{\langle p,s \rangle}(c_2) < \cdots < \eta_{\langle p,s \rangle}(c_{|R_{\langle p,s \rangle}|})$, that is, the generic clock $c_i$ holds the $i$-th smaller value in $R_{\langle p,s \rangle}$. A generic clock $c_i$ is said to be *irrelevant* to $\eta_{\langle p,s \rangle}$ if $\eta_{\langle p,s \rangle}(c_i) = \uparrow$, and *relevant* otherwise. Note that at most one generic clock is irrelevant to a clock interpretation.

Given two composite pairs $\langle p, s \rangle$ and $\langle p', s' \rangle$, if $|R_{\langle p,s \rangle}| = |R_{\langle p',s' \rangle}|$, then we interpret the two sets $C_{\langle p,s \rangle}$ and $C_{\langle p',s' \rangle}$ as being the same set of generic clock variables. The function $\eta_{\langle p,s \rangle}$ induces, for each $\langle q, \nu \rangle \in p$, a function $\mu : X_1 \to C_{\langle p,s \rangle}$ that associates to each clock $x \in X_1$ the generic clock which holds the value $\nu(x)$, that is, $\mu(x) = \eta^{-1}_{\langle p,s \rangle}(\nu(x))$ if $\nu(x) \leq \alpha$ and $\mu(x) = \eta^{-1}_{\langle p,s \rangle}(\uparrow)$ otherwise. The generalized location $\langle q, \nu \rangle$ is, then, *represented* by a pair $\langle q, \mu \rangle$, which we call a *position* of $\mathcal{A}_1$. Note that two different generalized locations

[2] The degree of nondeterminism of a timed automaton $\langle \Sigma, Q, Q_0, X, T, F \rangle$ is the cardinality of the greatest set $E \subseteq T$ such that all transitions in $E$ originate in the same location, are on the same symbol, and the conjunction of their clock constraints can be satisfied.

can be associated to the same position. This is because all values greater than $\alpha$ are mapped to $\uparrow$. For a composite pair $\langle p,s\rangle$, we define the set of positions of $\mathcal{A}_1$ as $P_{\langle p,s\rangle} = \{\langle q,\mu\rangle \mid \langle q,\mu\rangle$ represents some $\langle q,\nu\rangle \in p\}$. Similarly, we define the set $S_{\langle p,s\rangle}$ with respect to $\mathcal{A}_2$.

**The Equivalence Relation.** Now we define the equivalence relation $\sim$ over the set of composite pairs (compare to [1]). Given a number $t \in \mathbb{R}$, $\lfloor t\rfloor$ denotes the greatest integer smaller than or equal to $t$, and $\mathrm{fr}(t) = t - \lfloor t\rfloor$ denotes the fractional part of $t$. Define $\langle p,s\rangle \sim \langle p',s'\rangle$ iff:

- *same set of generic clocks*: $C_{\langle p,s\rangle} = C_{\langle p',s'\rangle}$;
- *same sets of positions*: $P_{\langle p,s\rangle} = P_{\langle p',s'\rangle}$, and $S_{\langle p,s\rangle} = S_{\langle p',s'\rangle}$;
- *equivalent generic clocks interpretations*:
  - *irrelevant clock*: for each $x \in C_{\langle p,s\rangle}$, $\eta_{\langle p,s\rangle}(x) = \uparrow$ iff $\eta_{\langle p',s'\rangle}(x) = \uparrow$;
  - *relevant clocks*:
    - for each $x \in C_{\langle p,s\rangle}$ and relevant to $\eta_{\langle p,s\rangle}$, $\lfloor \eta_{\langle p,s\rangle}(x)\rfloor = \lfloor \eta_{\langle p',s'\rangle}(x)\rfloor$ and $\mathrm{fr}(\eta_{\langle p,s\rangle}(x)) = 0$ iff $\mathrm{fr}(\eta_{\langle p',s'\rangle}(x)) = 0$;
    - for each pair $x$ and $y$ in $C_{\langle p,s\rangle}$, both relevant to $\eta_{\langle p,s\rangle}$, $\mathrm{fr}(\eta_{\langle p,s\rangle}(x)) < \mathrm{fr}(\eta_{\langle p,s\rangle}(y))$ iff $\mathrm{fr}(\eta_{\langle p',s'\rangle}(x)) < \mathrm{fr}(\eta_{\langle p',s'\rangle}(y))$, and $\mathrm{fr}(\eta_{\langle p,s\rangle}(x)) = \mathrm{fr}(\eta_{\langle p,s\rangle}(y))$ iff $\mathrm{fr}(\eta_{\langle p',s'\rangle}(x)) = \mathrm{fr}(\eta_{\langle p',s'\rangle}(y))$.

The relation $\sim$ records, for each generic clock, the interval from $\{[0,0],(0,1),[1,1],(1,2),\ldots,[\alpha,\alpha],(\uparrow)\}$ where the clock is contained. Note that any two clocks in the same interval satisfy the same set of clock constraints. To correctly update this information, the relation also records the order of the fractional parts for the relevant clocks. Nothing is needed, however, for clocks whose values are greater than $\alpha$, since all of them satisfy the same set of clock constraints. We refer the reader to [1] for a detailed discussion about this equivalence relation. In the sequel, we write $[\langle p,s\rangle]$ for the equivalence class to which $\langle p,s\rangle$ belongs.

We finish this section noting that the number of equivalence classes of $\sim$ is not finite, since there is no bound on the number of generic clocks. However, it is important to note that the number of equivalence classes with at most $K$ generic clocks *is* finite. Let $V_K$ denote the set of all equivalence classes with exactly $K$ generic clocks. The following bound holds (compare to [1]):

$$|V_K| < 2^{|Q_1|K^{|X_1|}} \times 2^{|Q_2|K^{|X_2|}} \times (2\alpha+2)^K \times K! \ .$$

### 3.2 Time Successors and the Graph $G$

Let $\Pi_{\mathcal{P}}$ denote the set of all equivalence classes of the relation $\sim$ over $\mathcal{P}$. We define now the subset construction region graph $G = (V,E)$. Its vertex set $V$ is a subset of $\Pi_{\mathcal{P}} \times \{1,2\}$. The reason why we need two copies of each equivalence class will be clear soon. $G$ has a unique initial vertex $\langle v_0, 1\rangle$, and each edge is labelled with one symbol from $\Sigma \cup \{\triangleright\}$. The new symbol $\triangleright$ represents a passage of time. Any edge from a vertex $\langle -, 1\rangle$ goes to a vertex $\langle -, 2\rangle$, and it has label $\triangleright$. Any edge from a vertex $\langle -, 2\rangle$ goes to a vertex $\langle -, 1\rangle$, and it has label $a$, for some $a \in \Sigma$. Thus, the graph $G$ is bipartite. The edge relation is defined in such a way that the graph is "deterministic", in the following sense: let a *run* of $G$ be an infinite sequence of the form: $\langle v_0, 1\rangle \xrightarrow{\sigma_1} \langle v_1, 2\rangle \xrightarrow{\sigma_2} \langle v_2, 1\rangle \xrightarrow{\sigma_3} \cdots$, such that, for all $i \geq 0$, there is an edge from $\langle v_i, -\rangle$ to $\langle v_{i+1}, -\rangle$ with label $\sigma_{i+1}$. Given a timed word $\rho = (\sigma, \tau)$, there exists, at most, one run $\langle v_0, 1\rangle \xrightarrow{\triangleright} \langle v_1, 2\rangle \xrightarrow{\sigma_1} \langle v_2, 1\rangle \xrightarrow{\triangleright} \langle v_3, 2\rangle \xrightarrow{\sigma_2} \langle v_4, 1\rangle \xrightarrow{\triangleright} \cdots$ of $G$, such that, for every $i \geq 1$, the following holds: $[\langle p_{\varrho_i}, s_{\varrho_i}\rangle] = v_{2i}$, where $\varrho_i$ is the finite timed word $(\sigma_1, \tau_1) \to (\sigma_2, \tau_2) \to \cdots \to (\sigma_i, \tau_i)$. We refer to this run as the run of $G$ over the timed word $\rho$. The undecidability of the inclusion problem manifests itself in the fact that $G$ may be an infinite graph. In Section 4 we give some sufficient conditions for $G$ to be finite. Once $G$ is finite, we show how one can obtain two Büchi $\omega$-automata $\mathcal{B}_1$ and $\mathcal{B}_2$ such that $L(\mathcal{B}_1) \subseteq L(\mathcal{B}_2)$ iff $L(\mathcal{A}_1) \subseteq L(\mathcal{A}_2)$. Thus, the problem will be reduced, in this case, to language inclusion of $\omega$-automata.

The graph $G$ is constructed inductively, from the initial vertex, by the definition of the edge relation. As in [1], we use the convenient notion of a time successor of an equivalence class to define the edge relation. In order to obtain an effective computational procedure, instead, one should define a representation for the equivalence classes and define the edge relation directly between the vertices. This can certainly be done, although with the cost of considering many different cases in the definition of the edge relation.

**Time Successors.** Let $v$ be an equivalence class. Consider a composite pair $\langle p, s\rangle$ in $v$. Given $t \in \mathbb{R}$, let $\langle p, s\rangle + t$ denote the composite pair obtained from $\langle p, s\rangle$ by replacing every clock interpretation $\nu$ in $p$ or in $s$, by $\nu + t$. An equivalence class $v'$ is a *time successor* of $v$ iff, given $\langle p, s\rangle$ in $v$, $v' = [\langle p, s\rangle + t]$, for some $t \in \mathbb{R}$, $t > 0$. Any equivalence class $v$ has finitely many time successors, since the number of generic clocks in any time successor of $v$ is, clearly, smaller than or equal to the number of generic clocks in $v$. In particular, for any $t_1$ and $t_2$, both greater than $\alpha$, $[\langle p, s\rangle + t_1] = [\langle p, s\rangle + t_2]$, which is an equivalence class with only one irrelevant generic clock.

**The graph $G$.** The graph $G$ has a unique initial vertex $\langle v_0, 1\rangle$, where $v_0 = [\langle p_0, s_0\rangle]$ and $p_0 = \{\langle q, \nu_0\rangle \mid q \in Q_{01}\}$, and $\nu_0$ is the clock interpretation which maps each $x \in X_1$ to zero. The same definition applies to $s_0$ with respect to $\mathcal{A}_2$. Note that, in fact, $\langle p_0, s_0\rangle$ is the unique composite pair in $v_0$. A vertex $\langle v, 1\rangle$ has an edge with label $\triangleright$ to a vertex $\langle v', 2\rangle$ iff $v'$ is a time successor of $v$. A vertex $\langle v, 2\rangle$ has an edge with label $a \in \Sigma$ to a vertex $\langle v', 1\rangle$ iff:

- Given a composite pair $\langle p, s\rangle$ in $v$, the following conditions hold:

  1. Given a generalized location $\langle q, \nu\rangle$ in $p$, there is a transition $\langle q, -, a, -, \delta\rangle$ in $T_1$, such that $\nu$ satisfies $\delta$; and
  2. $v' = [\langle p', s'\rangle]$, where:
     - $p' = \{\langle q', \nu'\rangle \mid$ there is $\langle q, \nu\rangle$ in $p$, and $\langle q, q', a, \lambda, \delta\rangle$ in $T_1$, such that $\nu$ satisfies $\delta$, and for each $x \in X_1$, $\nu'(x) = 0$ if $x \in \lambda$, and $\nu'(x) = \nu(x)$ otherwise$\}$;

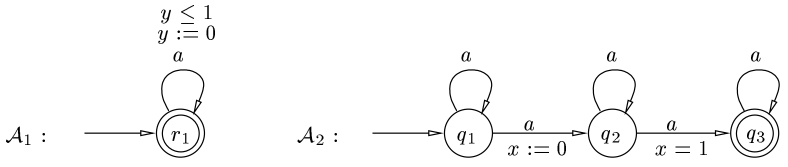

Figure 1: An instance for which $G$ is infinite

- $s' = \{\langle q', \nu' \rangle \mid$ there is $\langle q, \nu \rangle$ in $s$, and $\langle q, q', a, \lambda, \delta \rangle$ in $T_2$, such that $\nu$ satisfies $\delta$, and for each $x \in X_2$, $\nu'(x) = 0$ if $x \in \lambda$, and $\nu'(x) = \nu(x)$ otherwise$\}$.

Note that there is at most one edge out of a vertex $\langle v, 2 \rangle$ for each symbol in $\Sigma$, and that there may be a vertex $\langle v, 2 \rangle$ such that there is no edge out of $\langle v, 2 \rangle$. But, the initial vertex and condition 1. guarantee that for every vertex $\langle v, 1 \rangle \in V$, the $\mathcal{A}_1$-component is nonempty, that is, $p$ is nonempty for every composite pair $\langle p, s \rangle$ in $v$. On the other hand, the $\mathcal{A}_2$-component may be empty. We now give an example of an instance for which $G$ is infinite.

**Example of Infinite** $G$**.** Consider the instance in Fig. 1. The automaton $\mathcal{A}_2$ is the traditional example of a noncomplementable NTA [1]. Clearly, $L(\mathcal{A}_1) \not\subseteq L(\mathcal{A}_2)$. Consider the following finite timed words $\varrho_i = (a, 1 - 1/2^1) \to (a, 1 - 1/2^2) \to (a, 1 - 1/2^3) \to \cdots \to (a, 1 - 1/2^i)$, $i \geq 1$. The problem occurs because $\varrho_i$ has $i + 1$ different finite runs of $\mathcal{A}_2$ over it, and the clock $x$ is reset at distinct times for each run. Hence $D_{\langle p_{\varrho_i}, s_{\varrho_i} \rangle}$ has $i + 1$ distinct and relevant values, so that $[\langle p_{\varrho_i}, s_{\varrho_i} \rangle]$ has $i + 1$ generic clocks. For any $i \geq 1$, there is a timed word $\rho_i$, having $\varrho_i$ as a prefix and such that $\mathcal{A}_1$ has a run over $\rho_i$. Thus, if we try to construct $G$ we will need an infinite sequence of distinct vertices of the form: $\langle v_0, 1 \rangle \xrightarrow{\triangleright} \langle -, 2 \rangle \xrightarrow{a} \langle [\langle p_{\varrho 1}, s_{\varrho 1} \rangle], 1 \rangle \xrightarrow{\triangleright} \langle -, 2 \rangle \xrightarrow{a} \langle [\langle p_{\varrho 2}, s_{\varrho 2} \rangle], 1 \rangle \xrightarrow{\triangleright} \cdots$.

## 4 Progressive Automata

In the above example, the automaton $\mathcal{A}_1$ has runs over timed words where an arbitrary number of events can happen in a time interval of unit length. Allowing this property in the system model may drastically affect the complexity of the decision problems—it is an essential property in the proofs of the undecidability results about NTA [1, 5]. One may argue that a system model which can generate arbitrarily many discrete events in a finite interval of time is not realistic, since this is not physically realizable. In fact, many TA models of actual physical systems satisfy the property that there is a constant $K$ such that at most $K$ discrete events can happen, in any run, in a time interval of unit length. We say that these TA are *progressive*. For instance, in [3, 7] a TA model for asynchronous circuits is proposed. Every logical gate is followed by a delay element constraining, between lower and upper bounds, the rising and falling (which are the discrete events) of the digital signals. The lower bound is a positive constant, such that any cycle in the model takes at least $k$ time units to complete, for some positive constant $k$, and so, the automata are progressive;

see also [6] where this same discussion occurs in a similar formalism. It is worth noting that the progress requirement does not nullify the dense time assumption. In fact, one of the results in [3] is that cyclic circuits in that model, in general, do not admit discretization.

We can show that when the system model $\mathcal{A}_1$ is progressive, the graph $G$ is finite for any nondeterministic $\mathcal{A}_2$. Thus, one may use the full expressive power of NTA to specify real-time properties.

**Theorem 1** *If $\mathcal{A}_1$ is progressive, then $G$ is finite.*

**Proof 1 (Outline)** From the definitions, if there is an edge from a vertex $\langle v, 2\rangle$, where $v$ has $n$ generic clocks, to a vertex $\langle v', 1\rangle$, then $v'$ has, at most, $n+1$ generic clocks. Since the number of equivalence classes with at most $k$ generic clocks is finite, then $G$ is infinite iff, for every natural $k > 0$, $G$ has a vertex $\langle v, 1\rangle$, where $v$ has exactly $k$ generic clocks.

Let $N$ be a constant such that for every timed word that has a run of $\mathcal{A}_1$ over it, any sequence of $N$ events takes *more* than $\alpha$ time units. This constant exists since $\mathcal{A}_1$ is progressive. Assume that $G$ has a vertex $\langle v, 1\rangle$, where $v$ has $N+2$ generic clocks. By definition, there are $N+1$ relevant generic clocks in $v$. We can show, also from the definitions, that the relevant generic clock holding the greatest value in $v$ represents some clock, in some finite run of $\mathcal{A}_1$ or $\mathcal{A}_2$, that is not reset by the last $N$ transitions, for all finite timed words such that the run of $G$ over them ends in $\langle v, 1\rangle$. This is a contradiction to the fact that this generic clock is still relevant. □

We finish this section noting that the progressiveness of $\mathcal{A}_1$ is a sufficient but not necessary condition for the finiteness of $G$. As an example, consider the instance of Fig. 1. If we remove the command $x := 0$ from the transition from $q_1$ to $q_2$ in $\mathcal{A}_2$, the graph $G$ becomes finite, in spite of the non-progressiveness of $\mathcal{A}_1$. Another sufficient, although hardly acceptable, condition for the finiteness of $G$ is that every clock be reset, at least once, in every cycle of $\mathcal{A}_1$ and $\mathcal{A}_2$.

### 4.1 Obtaining the $\omega$-Automata.

In [1], given a timed automaton $\mathcal{A}$, an $\omega$-automaton $\mathcal{B}$ is obtained from the region graph $R(\mathcal{A})$, such that $L(\mathcal{A}) = \emptyset$ iff $L(\mathcal{B}) = \emptyset$. To this end, a correspondence between the runs of $\mathcal{A}$ and the runs of $R(\mathcal{A})$ is established through the concept of a *progressive* run of $R(\mathcal{A})$. This correspondence readily generalizes to our subset construction.

For a given equivalence class $v$, we define the set $P_v$ of positions of $\mathcal{A}_1$ as being equal to the set $P_{\langle p,s\rangle}$ for some composite pair $\langle p, s\rangle \in v$ (by definition, the set of positions of $\mathcal{A}_1$ is the same for every composite pair in $v$). The same definition holds for $S_v$, with respect to $\mathcal{A}_2$. Consider an edge from a vertex $\langle v, 2\rangle$ to a vertex $\langle v', 1\rangle$ with label $a$, for some $a \in \Sigma$. This edge naturally induces a relation between the positions in $P_v$ and the positions in $P_{v'}$. We say that a position $\langle q, \mu\rangle$ in $P_v$ is *a-linked* to a position $\langle q', \mu'\rangle$ in $P_{v'}$ iff:

- $\langle q, \mu\rangle$ represents some generalized location $\langle q, \nu\rangle$, and $\langle q', \mu'\rangle$ represents some generalized location $\langle q', \nu'\rangle$; and there is a transition $\langle q, q', a, \lambda, \delta\rangle$ in $T_1$, such that: $\nu$ satisfies $\delta$, and for each $x \in X_1$, $\nu'(x) = 0$ if $x \in \lambda$, and $\nu'(x) = \nu(x)$ otherwise.

Also, consider an edge from a vertex $\langle v, 1\rangle$ to a vertex $\langle v', 2\rangle$ with label $\triangleright$. We say that a position $\langle q, \mu\rangle$ in $P_v$ is $\triangleright$-*linked* to a position $\langle q', \mu'\rangle$ in $P_{v'}$ iff:

- $\langle q, \mu\rangle$ represents some generalized location $\langle q, \nu\rangle$, and $\langle q', \mu'\rangle$ represents some generalized location $\langle q', \nu'\rangle$; and, for some $t \in \mathbb{R}$, $t > 0$, $\langle q', \nu'\rangle = \langle q, \nu\rangle + t$.

Analogously, we can define the "linked" relation between the positions in $S_v$ and $S'_v$, with respect to $\mathcal{A}_2$. Given a run $r = \langle v_0, 1\rangle \xrightarrow{\sigma_1} \langle v_1, 2\rangle \xrightarrow{\sigma_2} \langle v_2, 1\rangle \xrightarrow{\sigma_3} \cdots$ of $G$, let an $\mathcal{A}_1$-*run* of $r$ be an infinite sequence $m_0 m_1 m_2 \cdots$ of positions of $\mathcal{A}_1$, such that, for all $i \geq 0$, $m_i \in P_{v_i}$, and $m_i$ is $\sigma_{i+1}$-linked to $m_{i+1}$. In a position $m = \langle q, \mu\rangle$, the function $\mu$ maps each clock $x$ in $X_1$ to a generic clock $c_x$ which is contained in exactly one interval from $\{[0,0], (0,1), [1,1], (1,2), \ldots, [\alpha, \alpha], (\uparrow)\}$. We already known, by the definition of $G$, that every timed word that has some run of $\mathcal{A}_1$ over it, has a run of $G$ over it. In a timed word $\rho$, time diverges, so that if $r$ is the run of $G$ over $\rho$, then, in every $\mathcal{A}_1$-run of $r$, every clock $x$ in $X_1$ is either reset (mapped to $[0,0]$) infinitely often, or, after some time, it increases without bound (is continuously mapped to $(\uparrow)$) [1]. Such $\mathcal{A}_1$-runs are called *progressive.* Call a run of $G$ progressive iff it has a progressive $\mathcal{A}_1$-run. The correspondence states that, for every progressive run $r$ of $G$, we can find a timed word $\rho$ such that $r$ is the run of $G$ over $\rho$ [1]. Then, clearly, either all $\mathcal{A}_1$-runs of $r$ are progressive, or none is progressive.

We treat first the case where $\mathcal{A}_1$ is known, in advance, to be progressive. In this case, all runs of $G$ are progressive. Afterwards, we discuss how to treat any $\mathcal{A}_1$. Assume that $\mathcal{A}_1$ is progressive. Given a run $r$ of $G$, consider the set $R_r$ of timed words $R_r = \{\rho \,|\, r \text{ is the run of } G \text{ over } \rho\}$. We have $|R_r| \geq 1$ (by the above discussion). We say that a position $\langle q, \mu\rangle$ is $\mathcal{A}_1$-*accepting* iff $q \in F_1$. A run $r = \langle v_0, 1\rangle \xrightarrow{\sigma_1} \langle v_1, 2\rangle \xrightarrow{\sigma_2} \langle v_2, 1\rangle \xrightarrow{\sigma_3} \cdots$ of $G$ is $\mathcal{A}_1$-*accepting* iff there is an $\mathcal{A}_1$-run $m_0 m_1 m_2 \cdots$ of $r$, such that for infinitely many $i \geq 0$, $m_i$ is $\mathcal{A}_1$-*accepting.* The same definitions hold with respect to $\mathcal{A}_2$. Then, either all timed words in $R_r$ are accepted by $\mathcal{A}_1$ or all timed words in $R_r$ are rejected by $\mathcal{A}_1$. The same holds for $\mathcal{A}_2$. Finally, the language inclusion problem reduces to verifying that, for every run $r$ of $G$, if $r$ is $\mathcal{A}_1$-accepting, then $r$ is $\mathcal{A}_2$-accepting.

We need to cope with a known difficulty of applying a subset construction to a Büchi automaton. Consider a run $r = \langle v_0, 1\rangle \xrightarrow{\sigma_1} \langle v_1, 2\rangle \xrightarrow{\sigma_2} \langle v_2, 1\rangle \xrightarrow{\sigma_3} \cdots$ of $G$. We *cannot* say that $r$ is $\mathcal{A}_1$-accepting if, for infinitely many $i \geq 0$, there is a $\mathcal{A}_1$-accepting position in $P_{v_i}$. We refer the reader to [9] for a solution to this difficulty, in the context of $\omega$-automata. As we will see, instead of trying to solve this directly on $G$, we will consider a nondeterministic image[3] of $G$ (somehow undoing the subset construction), so that the obtained $\omega$-automata will be nondeterministic and the mentioned difficulty is deferred to the algorithms for the language inclusion problem for nondeterministic $\omega$-automata.

**$\omega$-Automata.** A *Büchi $\omega$-automaton $\mathcal{B}$* is a tuple $\langle \Delta, Q, Q_0, T, F\rangle$, where

- $\Delta$ is a finite alphabet of symbols;

- $Q$ is a finite set of states;

[3] This is inspired by the deterministic image of a nondeterministic automaton [4].

- $Q_0 \subseteq Q$ is a set of initial states;
- $T \subseteq Q \times Q \times \Delta$ is a set of transitions;
- $F \subseteq Q$ is a set of final states.

A run $r$ of $\mathcal{B}$, over an $\omega$-word $\sigma = \sigma_1\sigma_2\cdots$, is an infinite sequence $r_0r_1r_2\cdots$ of states, such that $r_0 \in Q_0$, and for all $i \geq 1$, $\langle r_{i-1}, r_i, \sigma_i\rangle \in T$. A run $r$ is said to be *accepting* iff, for infinitely many $i \geq 1$, $r_i \in F$. The automaton $\mathcal{B}$ accepts an $\omega$-word $\sigma$ iff there is an accepting run of $\mathcal{B}$ over $\sigma$. The set $L(\mathcal{B})$ of $\omega$-words accepted by $\mathcal{B}$ is its *language.*

**The Nondeterministic Image.** The nondeterministic image encodes the runs of $G$ in the alphabet of the $\omega$-automata. Let $P = \{m \,|\, m \in P_v \text{ for some } \langle v, -\rangle \in V\}$ be the set of all positions of $\mathcal{A}_1$ in $G$. Then, $\mathcal{B}_1 = \langle \Delta, Q_1, Q_{01}, T_1, F_1\rangle$ and $\mathcal{B}_2 = \langle \Delta, Q_2, Q_{02}, T_2, F_2\rangle$, where $\Delta = \{\Sigma \cup \{\triangleright\}\} \times V$. For $\mathcal{B}_1$, we have: $Q_1 = V \times P$; $Q_{01} = \{\langle\langle v_0, 1\rangle, m\rangle \,|\, m \in P_{v_0}\}$; $T_1 = \{\langle\langle\langle v, i\rangle, m\rangle, \langle\langle v', j\rangle, m'\rangle, \langle \sigma, \langle v', j\rangle\rangle\rangle \,|\, m \in P_v, m' \in P_{v'}$, there is an edge from $\langle v, i\rangle$ to $\langle v', j\rangle$ with label $\sigma$, and $m$ is $\sigma$ -linked to $m'\}$; $F_1 = \{\langle\langle v, 1\rangle, m\rangle \,|\, m \text{ is } \mathcal{A}_1\text{-accepting}\}$. For $\mathcal{B}_2$, the same definitions hold, changing $S$ for $P$, $\mathcal{A}_2$ for $\mathcal{A}_1$, and $S_v$ for $P_v$. Then, the following theorem holds. Its proof is based on the correspondence between runs of $\mathcal{A}_1$ and $\mathcal{A}_2$, and runs of $G$; and on the correspondence between runs of $G$ and runs of $\mathcal{B}_1$ and $\mathcal{B}_2$:

**Theorem 2** *$L(\mathcal{A}_1) \subseteq L(\mathcal{A}_2)$ iff $L(\mathcal{B}_1) \subseteq L(\mathcal{B}_2)$.* □

**Non-progressive $\mathcal{A}_1$.** In this case, there may be a run $r$ of $G$, such that $|R_r| = 0$. The definition of acceptance for runs of $G$ need to be changed. Now, a run $r = \langle v_0, 1\rangle \xrightarrow{\sigma_1} \langle v_1, 2\rangle \xrightarrow{\sigma_2} \langle v_2, 1\rangle \xrightarrow{\sigma_3} \cdots$ of $G$ is $\mathcal{A}_1$-accepting iff there is an $\mathcal{A}_1$-run $m_0m_1m_2\cdots$ of $r$, such that for infinitely many $i \geq 0$, $m_i$ is $\mathcal{A}_1$-accepting; *and* for every clock $x \in X_1$, for infinitely many $i \geq 0$, $m_i = \langle q_i, \mu_i\rangle$ and $\mu_i$ maps $x$ to $[0, 0]$ or to $(\uparrow)$.

From the automaton $\mathcal{B}_1$, one can easily obtain an new automaton $\mathcal{C}_1$ that accounts for this new condition, in a standard way. If $X_1 = \{x_1, x_2, \ldots, x_n\}$, then $\mathcal{C}_1$ is made of $n + 1$ copies of $\mathcal{B}_1$. The new state space is $V \times P \times \{1, 2, \ldots, n + 1\}$. While reading an $\omega$-word, the control starts in the first copy, and jumps to the second copy as soon as it gets in a state $\langle\langle v, 1\rangle, \langle q, \mu\rangle, 1\rangle$, where $\mu$ maps $x_1$ to $[0, 0]$ or to $(\uparrow)$. This process repeats for every clock until the control reaches the last copy. Then it jumps back to the first copy, as soon as it gets in a state $\langle\langle v, 1\rangle, m, n + 1\rangle$, where $m$ is $\mathcal{A}_1$-accepting. The new set of final states is $\{\langle\langle v, 1\rangle, m, n + 1\rangle \,|\, m \text{ is } \mathcal{A}_1\text{-accepting}\}$.

## 5 Effective Infinite-State $\omega$-Automata

The semi-decision procedure presented above can be viewed as a reduction of the language inclusion problem for NTA to the language inclusion problem for nondeterministic effective infinite-state $\omega$-automata [10]. An infinite-state $\omega$-automaton $\mathcal{B} = \langle \Delta, Q, Q_0, T, F\rangle$ is *effective* if the sets $\Delta, Q, Q_0, T$ and $F$ are all recursive. That is, the sets may be infinite, but they are enumerable, and there is a Turing Machine $\mathcal{M}_{\mathcal{B}}$ that takes as input four integers

$w$, $x$, $y$ and $z$, always halts, and accepts the input iff: $x \in \Delta$ if $w = 1$; $x \in Q$ if $w = 2$; $x \in Q_0$ if $w = 3$; $\langle x, y, z \rangle \in T$ if $w = 4$; and $x \in F$ if $w = 5$ [10]. The machine $\mathcal{M}_{\mathcal{B}}$ can itself be encoded as an integer $\#\mathcal{B}$, which is called the *index* of $\mathcal{B}$.

Given any two NTA $\mathcal{A}_1$ and $\mathcal{A}_2$, we can easily derive, from the semi-decision procedure, two indexes $\#\mathcal{B}_1$ and $\#\mathcal{B}_2$, for two nondeterministic effective infinite-state $\omega$-automata $\mathcal{B}_1$ and $\mathcal{B}_2$, respectively, such that $L(\mathcal{A}_1) \subseteq L(\mathcal{A}_2)$ iff $L(\mathcal{B}_1) \subseteq L(\mathcal{B}_2)$. This gives a pleasant[4] way to show that language inclusion problem and the universality problem for NTA are in $\Pi^1_2$, since the language inclusion problem for nondeterministic effective infinite-state $\omega$-automata is known to be $\Pi^1_2$-complete [10] (we refer the reader to [8] for an introduction to the analytical hierarchy). The current complexity lower bound for the language inclusion problem and the universality problem for NTA is $\Pi^1_1$-hard [1]. The exact position of these undecidable problems in the analytical hierarchy is an interesting open problem.

## 6 Conclusions

In this paper, we presented a generalization of the region graph for NTA, which leads to a semi-decision procedure for testing language inclusion of NTA. We showed that TA models of real-time systems, satisfying a progress requirement, can be tested against any NTA. Interestingly enough, the semi-decision procedure is also a reduction of the language inclusion problem for NTA to the language inclusion problem for nondeterministic effective infinite-state $\omega$-automata.

The method *is*, as one should expect, extremely expensive from a practical point of view. The procedure reduces the language inclusion problem for NTA to the language inclusion problem for nondeterministic $\omega$-automata. The size of the latter problem is doubly exponential in the number of clocks, and exponential in the number of locations of the TA involved. Thus, one direction for future work is the development of heuristic methods and symbolic techniques for the problem.

Another direction is the interesting theoretical question about the exact position of the language inclusion problem and the universality problem for NTA in the analytical hierarchy. They are $\Pi^1_1$-hard [1], and belong to $\Pi^1_2$ (as a corollary of Section 5).

## References

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

[2] R. Alur, L. Fix, and T. Henzinger. Event-clock automata: A determinizable class of timed automata. In *CAV'94*, number 818 in Lecture Notes in Computer Science, pages 1–13, 1994.

[4] It is not difficult to show this result directly, since for every timed word we can find another timed word with rational occurrence times [1], which is equivalent for the purpose of language inclusion, such that timed words can be "paired" with functions from $\mathbb{N}$ to $\mathbb{N}$.

[3] E. Asarin, O. Maler, and A. Pnueli. On discretization of delays in timed automata and digital circuits. In *CONCUR'98*, number 1466 in Lecture Notes in Computer Science, pages 470–484, 1998.

[4] Y. Choueka. Theories of automata on $\omega$-tapes: A simplified approach. *Journal of Computer and System Sciences*, 8:117–141, 1974.

[5] T. Henzinger and J. Raskin. Robust undecidability of timed and hybrid systems. Technical Report USB/CSD-99-1073, Berkeley Univ., October 1999.

[6] H. Lewis. Finite-state analysis of asynchronous circuits with bounded temporal uncertainty. Technical Report TR-15-89, Harvard Univ., 1989.

[7] O. Maler and A. Pnueli. Timing analysis of asynchronous circuits using timed automata. In *CHARME'95*, number 987 in Lecture Notes in Computer Science, pages 189–205, 1995.

[8] H. Rogers. *Theory of Recursive Functions and Effective Computability.* McGraw-Hill, 1967.

[9] S. Safra. On the complexity of $\omega$-automata. In *Proc. of the 29th IEEE Foundations of Computer Science*, pages 319–327, October 1988.

[10] A. P. Sistla. On verifying that a concurrent program satisfies a nondeterministic specification. *Information Processing Letters*, 32:17–23, 1989.