Verification of Context-Free Timed Systems Using Linear Hybrid Observers*

Ahmed Bouajjani** † Rachid Echahed*** † Riadh Robbana** †

Abstract. We address the verification problem of infinite timed systems. We consider context-free timed systems defined as a generalization of the (regular) timed graphs [ACD90]. Then, we propose decision procedures for the verification of invariance properties of these systems, expressed by means of observation variables. These variables record relevant informations about the computations of the observed system. They are permanently updated along these computations without any interference with the behaviour of the system. Observation variables are either additional clocks (timers), nonbounded integer variables (accumulators), or constant slope continuous (real valued) variables (integrators).

1 Introduction

During the last decade, the framework of specification and verification of systems has been extensively developed. The first works in this area consider the case of systems modeled by *finite-state* automata. For such systems, several specification and verification methods have been elaborated, using behavioural equivalence (or preorder) testing [Mil80, Par81, KS83] or model-checking techniques for logic-based specifications [QS82, CES83, VW86, EL86, CS91]. Recent investigations aim at extending these techniques in order to deal with two kinds of systems: *timed* and/or *hybrid systems*, in particular with a dense time domain (positive reals) [ACD90, Cer92, HNSY92, ACHH93, BES93, KPSY93], and separately, systems having (discrete) infinite state spaces, in particular *context-free systems* (or equivalently systems supplied with a stack) [BBK87, BS92, CHS92]. This paper presents a unification of these two directions of investigations, by proposing automatic verification methods for *context-free timed systems*. First, let us give a brief overview of the existing works on the verification of timed systems and context-free systems.

Concerning timed systems, several models and description formalisms have been proposed [ACD90, NRSV90, Wan90, CGL93]. One of the most adopted models for these systems is certainly the powerful model of finite *timed graphs* introduced in [ACD90]. Model-checking algorithms have been proposed for timed graphs w.r.t. formulas of the timed temporal logic TCTL [ACD90, HNSY92]. Recently, the scope of the investigations around timed systems has been broadened by considering specification languages allowing to reason about new interesting notions as *durations*, and by extending the models and the verification techniques for timed systems to the case of hybrid systems

* Partially supported by the ESPRIT-BRA project REACT.

** VERIMAG-SPECTRE, Miniparc-Zirst, Rue Lavoisier, 38330 Montbonnot St-Martin, France. VERIMAG is a joint laboratory of CNRS, INPG, UJF and VERILOG SA., SPECTRE is a project of INRIA.

*** LGI-IMAG, BP53, 38041 Grenoble cedex, France.

† Ahmed.Bouajjani@imag.fr, Rachid.Echahed@imag.fr, Riadh.Robbana@imag.fr

[CHR91, MMP92, MP93, ACHH93, ACH93, BES93, KPSY93, NOSY93]. The notion of duration of a state property has been introduced in [CHR91] and corresponds to the accumulated time spent in some given computation sequence while the considered property holds. So, the notion of duration allows to reason about several times corresponding to durations of some relevant properties. For instance, we may require in the specification of a communication node that

1. *in every session, the time consecrated to recover lost messages is less than 10 % of the global time.*

Some automatic verification techniques have been proposed in order to decide whether a finite timed system meets some kinds of duration properties as invariance in [KPSY93, ACH93] and nested invariance and eventuality properties in [BES93]. These techniques allow to verify for instance the property (1) above.

Concerning the works on context-free systems, important results have been obtained in the last few years about the (un)decidability of behavioural equivalences and preorders [BBK87, GH91, CHS92]. Mainly, it has been shown that bisimulation equivalence is decidable for these systems [CHS92]. There are less works concerning the extension of the logic-based specification and verification approach to the case of context-free systems. The first work on this topic proposes a model-checking algorithm for *regular* properties of these systems, expressed in the propositional μ-calculus [BS92]. However, a wide class of relevant properties of infinite-state systems (in particular, the context-free ones) are nonregular properties and cannot be expressed in the μ-calculus neither by finite-state ω-automata. For instance, in the specification of a communication node, we may need to express the fact that

2. *between the beginning and the ending of every session, there are exactly the same numbers of receptions and transmissions.*

In addition, we have also to require that

3. *during every session, the number of transmissions never exceeds the number of receptions.*

Actually, as these examples show, significant properties of infinite-state systems are essentially properties involving constraints on *numbers of occurrences* of some events (or number of states satisfying some state property). In [BER93] we have proposed a temporal logic that allows to express such constraints using formulas in Presburger arithmetic, and we have proposed automatic verification procedures for a wide class of nonregular properties, including (2) and (3), for (untimed) context-free systems.

In this paper, we address the verification problem for *context-free timed systems*, i.e., timed systems such that the underlying untimed structure is a context-free system. First, we propose a definition of these systems which is a generalization of the definition of (finite) timed graphs given in [ACD90]. Then, we consider the verification problem of context-free timed systems w.r.t. properties involving constraints on delays, durations, and on numbers of occurrences of some (events) state properties.

The properties we tackle in this paper are *invariance properties*, and our approach of verification is based on the use of *linear hybrid observers*. In general, given a system to verify, an observer is another system running in parallel, which maintains a set of *observation variables*, where informations about the execution of the observed system

are recorded. The observer does not interfere with the execution of the system and its aim is only to check permanently the truth of some invariance property on its observation variables.

We define a linear hybrid observer as an observer that disposes of three kinds of observation variables: timers that can be reset at some transitions of the observed system, discrete (integer valued) variables, called *accumulators*, that can be incremented or decremented at some transitions of the system, and finally, continuous (real valued) variables, called *integrators*, that change continuously with a constant slope (which may be different) at each location of the observed system. The accumulators allow to count linear combinations of numbers of occurrences of some events whereas the integrators allow to count linear combinations of durations.

The remainder of this paper is organized as follows. In Section 2, we introduce some notations about sequences and context-free grammars. In Section 3, we define context-free timed systems, in Section 4, we define their linear hybrid observers. In Section 5, we define *invariance formulas* and dually *reachability formulas*, expressed on observation variables of linear hybrid observers. In Section 6, we present decidability results for the verification problem of such invariance formulas on context-free timed systems. For this, we consider two different cases: The first case (see Section 6.1) corresponds to the consideration of pure timed constraints, i.e., the observer has only timers and no accumulators nor integrators. We give for this case a decision procedure for the verification of invariance formulas based on a reduction to the emptiness problem of context-free languages. In the second case (see Section 6.2) we prove the decidability of the verification problem of reachability formulas that involve constraints on timers and accumulators but at most one integration constraint. This result is proven for CFTS's whose transition guards are non-strict, by reducing the considered problem to the satisfiability problem of integer linear constraints.

2 Preliminaries

We introduce in this section some notations and recall some well-known notions about sequences, languages and grammars.

Let Σ be a finite alphabet. We denote by Σ^* the set of finite sequences over Σ. Given a sequence $\sigma \in \Sigma^*$, $|\sigma|$ denotes the length of σ. Let λ denote the empty sequence, i.e., the sequence of lenght 0. Let $\Sigma^+ = \Sigma - \{\lambda\}$. For every $a \in \Sigma$, $|\sigma|_a$ is the number of occurrences of a in σ. In the sequel, we write $a \in \sigma$ to denote the fact that a appears in the sequence σ. For every $i \in \{1, \ldots, |\sigma|\}$, $\sigma(i)$ is the i^{th} element of σ. Consider a nonempty sequence $\sigma \in \Sigma^*$. For every $i \in \{1, \ldots, |\sigma|\}$, we denote by σ_i the prefix of σ ending at position i, i.e., the subsequence $\sigma(1) \cdots \sigma(i)$. For $i = 0$, we consider that $\sigma_i = \lambda$.

A context-free grammar G over Σ is a tuple $(\Sigma, N, Prod, S)$ where N is a set of nonterminals, $Prod$ is a set of productions of the form $A \rightarrow \alpha$ where $A \in N$ and $\alpha \in (\Sigma \cup N)^*$, and S is the starting symbol. Given a production $p =$ "$A \rightarrow \alpha$" $\in Prod$, we denote by $lhs(p)$ the left hand side of p (i.e., A) and by $rhs(p)$ its right hand side (i.e., α). We adopt standard notations for the derivation relation ($\Longrightarrow$) and its reflexive-transitive closure ($\overset{*}{\Longrightarrow}$). We use subscripts to precise the set of productions or the sequences of productions used in the derivation. We denote by $L(G)$ the language generated by the grammar G (i.e., the set of sequences $\sigma \in \Sigma^*$ such that $S \overset{*}{\Longrightarrow} \sigma$). We use CFG to

abbreviate *context-free grammar* and CFL for *context-free language*. For more details concerning the theory of formal languages, see for instance [Har78].

3 Context-Free Timed Systems

Finite timed graphs have been introduced in [ACD90] as a powerful model for real-time systems. A timed graph consists in a finite location graph augmented by a set of timers (clocks) that can be reset and/or tested at each transition. A computation of the so modeled system can visit some location and stay in it by letting time progress (during the visit, the timers run continuously), and then moves to another location by taking some transition in the graph, provided that the enabling guard (condition on the values of the timers) is satisfied ; some of the timers may be reset after taking the transition.

In this paper, our aim is to deal with a class of infinite timed graphs. We consider timed graphs having a context-free structure, i.e., they are defined by a context-free set of rules (similar to productions in CFG's). Actually, we generalize the definition of timed graphs by replacing their binary transition relation between locations with a relation between locations and sequences of locations. Let us now give the formal definitions.

Let $\mathcal{P}rop$ be a set of atomic propositions. Let $\mathcal{C}$ be a set of timers (clocks). A *time guard* on $\mathcal{C}$ is any boolean combination of constraints of the form $x \prec n$ where $x \in \mathcal{C}$, $\prec \in \{<, \leq\}$ and $n \in I\!N$. Let $\Gamma_{\mathcal{C}}$ be the set of time guards on $\mathcal{C}$.

A *context-free timed system* (CFTS for short) is a tuple $\mathcal{M} = (\Sigma, \mathcal{V}ar, \mathcal{C}, \delta, \gamma, \rho, \Pi)$ where $\Sigma = 2^{\mathcal{P}rop}$, $\mathcal{V}ar$ is a set of location variables, $\mathcal{C}$ is a set of timers, $\delta \subseteq \mathcal{V}ar \times \mathcal{V}ar^*$ is a set of derivation rules, $\gamma : \delta \rightarrow \Gamma_{\mathcal{C}}$, $\rho : \delta \rightarrow 2^{\mathcal{C}}$ and $\Pi : \mathcal{V}ar \rightarrow \Sigma$. The function γ associates with each derivation rule $d \in \delta$ a time guard that should be satisfied by the values of the timers when d is applied, the function ρ associates with each rule d the set of timers that should be reset at each application of d. Finally, the function Π associates with each location variable $X \in \mathcal{V}ar$ the set of atomic propositions that hold at X.

An alternative definition of context-free timed systems can be given using pushdown automata instead of sets of derivation rules. We can translate one definition to the other by extending the standard transformations between context-free grammars and pushdown automata. Notice that a particular case of such systems are 1-counter systems, i.e., systems with one nonbounded integer variable.

Now, we give the *operational semantics* of the CFTS $\mathcal{M}$. A state of the system $\mathcal{M}$ is constituted by a *nonempty* sequence of location variables and a valuation that assigns to each timer a positive real value, i.e., a state is a pair $\langle \alpha, E \rangle$ where $\alpha \in \mathcal{V}ar^+$ and $E : \mathcal{C} \rightarrow I\!R^+$. Let $\mathcal{S}_{\mathcal{M}}$ be the set of states of the system $\mathcal{M}$.

We define two transition relations $\rightarrow$ and $\triangleright$ between states of $\mathcal{M}$. The relation $\rightarrow$ corresponds to transitions due to time progress at some location whereas $\triangleright$ corresponds to moves between locations using derivation rules in δ. Before giving the formal definition of these relations, let us first introduce some notations.

Given a derivation rule $d = (X, \alpha)$, we denote by $fst(d)$ (resp. $snd(d)$) the first (resp. second) component of the rule, i.e. the location variable X (resp. the sequences of location variables α). Given a valuation $E : \mathcal{C} \rightarrow I\!R^+$, a timer $x \in \mathcal{C}$ and a time value $t \in I\!R^+$, we denote by $E[x \leftarrow t]$ the new valuation which assigns t to x and coincides with E for all the other timers. Moreover, for any $t \in I\!R^+$, we denote by $E + t$ the valuation E' such that for every $x \in \mathcal{C}$, $E'(x) = E(x) + t$. Finally, given a valuation E and a constraint g, we denote by $E \vdash g$ the fact that the evaluation of g under the valuation E is true.

Now, we define two families of relations $\xrightarrow{t}$ and $\rhd_d$ with $t \in \mathbb{R}^+$ and $d \in \delta$. For every $t \in \mathbb{R}^+$ and every $d \in \delta$, these relations are defined as the smallest relations included in $\mathcal{S}_\mathcal{M} \times \mathcal{S}_\mathcal{M}$ such that:

- $\langle \alpha, E \rangle \xrightarrow{t} \langle \alpha, E + t \rangle$,
- $fst(d) = X$ and $E \vdash \gamma(d)$ implies $\langle X \cdot \alpha, E \rangle \rhd_d \langle snd(d) \cdot \alpha, E[x \leftarrow 0]_{x \in \rho(d)} \rangle$

Notice that in the definition of $\rhd_d$, we suppose that $snd(d) \cdot \alpha \neq \lambda$. Nevertheless, this assumption is not a restriction, since in case λ is derivable via the relation δ from some $\beta \in \mathcal{V}ar^+$, we can add to the system a fresh variable $\sharp$ which represents a termination location, and for any state $\langle \beta, E \rangle$, consider rather the state $\langle \beta \cdot \sharp, E \rangle$.

We define $\rightarrow = \bigcup_{t \geq 0} \xrightarrow{t}$ and $\rhd = \bigcup_{d \in \delta} \rhd_d$, and we consider the relation $\hookrightarrow = \rightarrow \cup \rhd$. We denote by $\overset{*}{\hookrightarrow}$ the reflexive-transitive closure of $\hookrightarrow$. For any pair of states s and s', $s \overset{*}{\hookrightarrow} s'$ means that s' is *reachable* from s in $\mathcal{M}$.

Given a state s, a *computation sequence* of $\mathcal{M}$ starting from s is a sequence

$$\langle s_0, \tau_0 \rangle \langle s_1, \tau_1 \rangle \cdots \langle s_n, \tau_n \rangle \in \mathcal{S}_\mathcal{M}^*$$

such that $s_0 = s$, $\tau_0 = 0$, and for every i such that $0 \leq i < n$, either $s_i \rhd s_{i+1}$ and $\tau_i = \tau_{i+1}$, or there exists an amount of time $t \in \mathbb{R}^+$ such that $s_i \xrightarrow{t} s_{i+1}$ and $\tau_{i+1} = \tau_i + t$. We denote by $Comput(\mathcal{M}, s)$ the set of computation sequences of $\mathcal{M}$ starting from s.

4 Linear Hybrid Observers

Let $\mathcal{M} = (\Sigma, \mathcal{V}ar, \mathcal{C}, \delta, \gamma, \rho, \Pi)$ be a CFTS. A *linear hybrid observer* (LHO) for the system $\mathcal{M}$ is a tuple $\mathcal{O} = (\mathcal{X}, \mathcal{A}, \mathcal{I}, \zeta, \kappa, \partial)$ where $\mathcal{X}$ is a set of timers, $\mathcal{A}$ is a set of accumulators (discrete integer variables), $\mathcal{I}$ is a set of integrators (constant slope continuous real variables), $\zeta : \delta \rightarrow 2^\mathcal{X}$, $\kappa : \delta \rightarrow (\mathcal{A} \rightarrow \mathbb{Z})$ and $\partial : \mathcal{V}ar \rightarrow (\mathcal{I} \rightarrow \mathbb{Z})$. The function ζ associates with each derivation rule $d \in \delta$ the set of timers in $\mathcal{X}$ that should be reset at each application of d. The function κ associates with each rule d and accumulator $u \in \mathcal{A}$, an integer that should be added to u whenever d is applied. Finally, the function ∂ associates with each location variable $X \in \mathcal{V}ar$ and integrator $u \in \mathcal{I}$, an integer rate at which u changes continuously while the computation is at X. This means that if the computation stays t amount of time at X, the variation of u is $\partial(X)(u) \cdot t$.

The composition of the system $\mathcal{M}$ with the observer $\mathcal{O}$ is a context-free hybrid system $\mathcal{M}_\mathcal{O}$. A state of this system is an enrichment of a state of $\mathcal{M}$ by valuations for the new timers, accumulators and integrators of $\mathcal{O}$. Let $\mathcal{T} = \mathcal{C} \cup \mathcal{X}$. Formally, a state of $\mathcal{M}_\mathcal{O}$ is a tuple $\langle \alpha, E_\mathcal{T}, E_\mathcal{A}, E_\mathcal{I} \rangle$ where $\alpha \in \mathcal{V}ar^+$, $E_\mathcal{T} : \mathcal{T} \rightarrow \mathbb{R}^+$, $E_\mathcal{A} : \mathcal{A} \rightarrow \mathbb{Z}$ and $E_\mathcal{I} : \mathcal{I} \rightarrow \mathbb{R}$. Let $\mathcal{S}_{\mathcal{M}_\mathcal{O}}$ be the set of states of the hybrid system $\mathcal{M}_\mathcal{O}$.

The computations of $\mathcal{M}_\mathcal{O}$ are obtained as well by enriching the computations of $\mathcal{M}$. We extend the definitions of the transition relations $\xrightarrow{t}$ and $\rhd_d$ to states of $\mathcal{M}_\mathcal{O}$. Let us first introduce some notations. Given a valuation of the accumulators $E : \mathcal{A} \rightarrow \mathbb{Z}$ and a mapping $D : \mathcal{A} \rightarrow \mathbb{Z}$, we denote by $E + D$ the new valuation E' such that for every $u \in \mathcal{A}$, $E'(u) = E(u) + D(u)$. Moreover, given a valuation of the integrators $E : \mathcal{I} \rightarrow \mathbb{R}$, a mapping $R : \mathcal{I} \rightarrow \mathbb{Z}$ and an amount of time $t \in \mathbb{R}^+$, we denote by $E + R \cdot t$ the new valuation E' such that, for every $u \in \mathcal{I}$, $E'(u) = E(u) + R(u) \cdot t$.

Now, for every amount of time $t \in \mathbb{R}^+$ and every derivation rule $d \in \delta$, we consider that $\xrightarrow{t}$ and $\rhd_d$ are the smallest relations included in $\mathcal{S}_{\mathcal{M}_\mathcal{O}} \times \mathcal{S}_{\mathcal{M}_\mathcal{O}}$ such that:

- $\langle X \cdot \alpha, E_{\mathcal{T}}, E_{\mathcal{A}}, E_{\mathcal{I}} \rangle \xrightarrow{t} \langle X \cdot \alpha, E_{\mathcal{T}} + t, E_{\mathcal{A}}, E_{\mathcal{I}} + \partial(X) \cdot t \rangle$,
- $fst(d) = X$ and $E \vdash \gamma(d)$ implies
  $\langle X \cdot \alpha, E_{\mathcal{T}}, E_{\mathcal{A}}, E_{\mathcal{I}} \rangle \rhd_d \langle snd(d) \cdot \alpha, E_{\mathcal{T}}[x \leftarrow 0]_{x \in \rho(d) \cup \zeta(d)}, E_{\mathcal{A}} + \kappa(d), E_{\mathcal{I}} \rangle$

We extend also the relations $\rightarrow$ and $\rhd$, as well as the relation $\hookrightarrow = \rightarrow \cup \rhd$, to states of $\mathcal{M}_{\mathcal{O}}$ and define the computation sequences of $\mathcal{M}_{\mathcal{O}}$ accordingly. Given a state $s \in \mathcal{S}_{\mathcal{M}_{\mathcal{O}}}$, we denote by $\mathcal{C}omput(\mathcal{M}_{\mathcal{O}}, s)$ the set of computation sequences of $\mathcal{M}_{\mathcal{O}}$ starting from s.

5 Invariance Properties

Given a CFTS $\mathcal{M}$ and a LHO $\mathcal{O}$, we consider the set of *invariance formulas* φ defined by:

$$\begin{array}{l} \varphi ::= \forall\Box \bigwedge \bigvee \psi \\ \psi ::= \pi \mid \xi \\ \pi ::= P \mid \neg\pi \mid \pi \vee \pi \\ \xi ::= u \sim k \quad \text{with } u \in \mathcal{X} \cup \mathcal{A} \cup \mathcal{I}, \sim \in \{<, >, \leq, \geq\}, k \in \mathbb{Z} \end{array}$$

A constraint $\xi = u \sim k$ is called either a *time*, *accumulation* or *integration* constraint according to the fact that u is respectively a timer, an accumulator or an integrator.

In order to give the semantics of invariance formulas, we define a satisfaction relation $\models$ between states of $\mathcal{M}_{\mathcal{O}}$ and these formulas.

Let $s = \langle \alpha, E_{\mathcal{T}}, E_{\mathcal{A}}, E_{\mathcal{I}} \rangle$ be a state of $\mathcal{M}_{\mathcal{O}}$. Suppose that the sets $\mathcal{T}$, $\mathcal{A}$ and $\mathcal{I}$ have no common elements and let E be the union of the functions $E_{\mathcal{T}}$, $E_{\mathcal{A}}$ and $E_{\mathcal{I}}$. The satisfaction relation $\models$ is inductively defined by:

$$\begin{array}{ll} s \models \forall\Box\phi & \text{iff } \forall s'.\ s \overset{*}{\hookrightarrow} s'.\ s' \models \phi \\ s \models P & \text{iff } P \in \Pi(\alpha(1)) \\ s \models \neg\phi & \text{iff } s \not\models \phi \\ s \models \phi_1 \vee \phi_2 & \text{iff } s \models \phi_1 \text{ or } s \models \phi_2 \\ s \models u \sim k & \text{iff } E \vdash u \sim k \end{array}$$

Let us introduce the operator $\exists\Diamond = \neg\forall\Box\neg$. It is easy to see from the definition of the relation $\models$ that $s \models \exists\Diamond\phi$ if and only if there exists some state s' that is reachable from s and satisfies ϕ. Formulas of the form $\exists\Diamond\phi$ are called *reachability formulas.*

Using standard laws for the boolean connectives together with the fact that any formula $\exists\Diamond(\phi_1 \vee \phi_2)$ is equivalent to $(\exists\Diamond\phi_1) \vee (\exists\Diamond\phi_2)$, it can easily be shown that every invariance formula is equivalent to the negation of a formula of the form

$$\bigvee_{i=1}^{n} \exists\Diamond(\pi_i \wedge \bigwedge_{j=1}^{m_i} \xi_i^j)$$

where we assume without loss of generality that if ξ_i^j is a time constraint, then it is of the form $u \sim n$ with $n \in \mathbb{N}$, if ξ_i^j is an accumulation constraint, then it is of the form $u \leq k$, and finally, if ξ_i^j is an integration constraint, it is of the form $u \prec k$ with $\prec \in \{<, \leq\}$.

6 Decidability Results

We present gradually decidability results concerning the the satisfaction relation $\models$ between states of CFTS's and invariance formulas. First, we consider the case when the

observer uses only timers (clocks). Thus, the invariance formulas in this case contain only (pure) time constraints and correspond to a subclass of TCTL formulas [ACD90]. Then, we consider the case when the observer uses in addition accumulators and integrators. The consideration of integration constraints imposes some restrictions as in [KPSY93].

Let us fix for the remainder of this section a CFTS $\mathcal{M} = (\Sigma, \mathcal{V}ar, \mathcal{C}, \delta, \gamma, \rho, \Pi)$ and a LHO $\mathcal{O} = (\mathcal{X}, \mathcal{A}, \mathcal{I}, \zeta, \kappa, \partial)$, and let $\mathcal{T} = \mathcal{C} \cup \mathcal{X}$. We suppose without loss of generality that the sets $\mathcal{C}$, $\mathcal{X}$, $\mathcal{A}$ and $\mathcal{I}$ are disjoint.

6.1 Pure Time Constraints

In this subsection we tackle the verification problem of invariance (resp. reachability) formulas involving only time constraints. We call these formulas pure time constraints (PTC) invariance (resp. reachability) formulas. Actually, we can consider equivalently that the observer $\mathcal{O}$ is such that $\mathcal{A} = \mathcal{I} = \emptyset$. We show that in this case, the verification problem of reachability formulas (hence, invariance formulas) is decidable by reducing it to the nonemptiness problem of context-free languages, using and extending the notion of *region graphs* introduced in [ACD90].

In the remainder of this subsection, valuations of accumulators and integrators are omitted in representations of states since they are undefined. So, we suppose that a state is a pair $\langle \alpha, E \rangle$ where E is a valuation of timers. Let $\mathcal{E} = [\mathcal{T} \rightarrow \mathbb{R}^+]$ be the set of valuations of timers. In [ACD90], an equivalence relation $\cong$ on $\mathcal{E}$ is defined so that any pair of states of a (finite) timed graph with equivalent time valuations satisfies the same reachability formulas (actually, the same TCTL formulas). It can be easily verified that this result holds also for context-free timed systems.

Lemma 6.1 *Let $\alpha \in \mathcal{V}\mathrm{ar}^+$ and φ be a PTC reachability formula. Then, $\forall E, E' \in \mathcal{E}$, $E \cong E'$ implies $\langle \alpha, E \rangle \models \varphi$ iff $\langle \alpha, E' \rangle \models \varphi$.*

The Lemma above allows to analyse the satisfaction of reachability formulas on a countable structure instead of the noncountable state graph $(\mathcal{S}_{\mathcal{M}_\mathcal{O}}, \hookrightarrow)$. This countable structure is what is called *region graph* in [ACD90]. It corresponds to the quotient graph of $(\mathcal{S}_{\mathcal{M}_\mathcal{O}}, \hookrightarrow)$ w.r.t. $\cong$. Let us see how this structure is obtained.

First of all, it is shown in [ACD90] that given a formula φ, the relation $\cong$ induces a *finite* partition of $\mathcal{E}$, depending on the constants that are compared with timers both in the time guards of $\mathcal{M}$ and in time constraints of φ. Then, consider a PTC reachability formula $\varphi = \exists\Diamond(\pi \wedge \bigwedge_{i=1}^{n} \xi_n)$, and let $[\mathcal{E}]$ be the quotient set of $\mathcal{E}$ under the relation $\cong$. We denote by $[E]$ the equivalence class of $E \in \mathcal{E}$.

Also, from [ACD90], the set $[\mathcal{E}]$ can be supplied by a successor function *succ* between equivalence classes which captures time progress. The function *succ* is defined in the following manner: For every $E, E' \in \mathcal{E}$, $succ([E]) = [E']$ if and only if $E \ncong E'$, and $\exists t \in \mathbb{R}^+$ such that $E' = E + t$, and $\forall t' \in \mathbb{R}^+$, $0 \leq t' < t$, either $E + t' \cong E$ or $E + t' \cong E'$.

Now, we define a *region* as a pair $\langle \alpha, [E] \rangle$, where $\alpha \in \mathcal{V}ar^+$ and $[E] \in [\mathcal{E}]$. Let *Reg* be the set of such regions. The region graph $\mathcal{R}(\mathcal{M}_\mathcal{O}, \varphi)$ defined by $\mathcal{M}_\mathcal{O}$ and φ, is defined as the graph $\langle Reg, Edg \rangle$ where *Reg* is the vertex set and the edge set *Edg* is the smallest set such that:

- Each vertex (region) $\langle \alpha, [E] \rangle$ has an edge to $\langle \alpha, succ([E]) \rangle$,

- Each vertex $\langle X \cdot \alpha, [E]\rangle$ has an edge to $\langle \beta \cdot \alpha, [E[x \leftarrow 0]_{x \in T}]\rangle$, for every rule $d = (X, \beta) \in \delta$ such that $E \vdash \gamma(d)$

The region graph can be seen as a kind of product of the context-free timed system $\mathcal{M}_\mathcal{O}$ and the finite state graph $([\mathcal{E}], succ)$. Notice that the region graphs considered in [ACD90] are finite since they are obtained from a finite (regular) timed graphs. Here, a region graph is in general infinite but has a *context-free structure.*

Now, let $s = \langle \alpha, E\rangle$ be a state and suppose that we are interested in the problem of checking whether $s \models \varphi$. By Lemma 6.1, this problem reduces to decide whether there exists some path in the region graph $\mathcal{R}(\mathcal{M}_\mathcal{O}, \varphi)$ which starts from $\langle \alpha, [E]\rangle$ and reaches some region $\langle \alpha', [E']\rangle$ such that $\langle \alpha', E'\rangle \models (\pi \wedge \bigwedge_{i=1}^{n} \xi_i)$. We show that this problem is actually decidable by reducing it to the nonemptiness problem of a context-free language $\mathcal{G}_{(s,\varphi)}$. This grammar has an empty alphabet (so, its language is either empty or equal to $\{\lambda\}$) and it is defined in such a manner that all its successful derivations correspond to paths in the region graph starting from $\langle \alpha, [E]\rangle$ and reaching some region $\langle \alpha', [E']\rangle$ such that $\langle \alpha', E'\rangle \models (\pi \wedge \bigwedge_{i=1}^{n} \xi_i)$.

In the sequel, given a location variable X and a state formula π, we write $\Pi(X) \models \pi$ if $(\bigwedge_{P \in \Pi(X)} P) \wedge (\bigwedge_{P \notin \Pi(X)} \neg P) \Rightarrow \pi$. Now, let $\mathcal{F} = \{[E] \in [\mathcal{E}] \ : \ E \vdash \bigwedge_{i=1}^{n} \xi_i\}$. We define the CFG $\mathcal{G}_{(s,\varphi)}$ by $(\emptyset, N, Prod, S)$ where $N = \{(\!|[F], X, [F']|\!) \ : \ X \in \mathcal{V}ar$ and $[F], [F'] \in [\mathcal{E}]\} \cup \{[\![[F], X, [F']]\!] \ : \ X \in \mathcal{V}ar$ and $[F], [F'] \in [\mathcal{E}]\} \cup \{S\}$ and the set of productions $Prod$ is the smallest set containing the following productions:

(P1) $S \rightarrow (\!|[E_0], \alpha(1), [E_1]|\!) \cdots (\!|[E_{i-1}], \alpha(i), [E_i]|\!)[\![[E_i], \alpha(i+1), [E_{i+1}]]\!]$
if $E_0 = E$, $i \in \{0, \ldots, |\alpha| - 1\}$, $\forall j \in \{1, \ldots, i\}, [E_j] \in [\mathcal{E}]$, and $[E_{i+1}] \in \mathcal{F}$,

(P2) $(\!|[E_1], X, [E_2]|\!) \rightarrow (\!|succ([E_1]), X, [E_2]|\!)$,

(P3) $(\!|[E_1], X, [E_2]|\!) \rightarrow$
$(\!|[F_0], \beta(1), [F_1]|\!) \cdots (\!|[F_{m-2}], \beta(m-1), [F_{m-1}]|\!)(\!|[F_{m-1}], \beta(m), [E_2]|\!)$
if $d = (X, \beta) \in \delta$ with $|\beta| = m > 0$, $E_1 \vdash \gamma(d)$,
$F_0 = E_1[x \leftarrow 0]_{x \in \rho(d) \cup \zeta(d)}$, and $[F_1], \ldots, [F_{m-1}] \in [\mathcal{E}]$,

(P4) $(\!|[E_1], X, [E_2]|\!) \rightarrow \lambda$
if $d = (X, \lambda) \in \delta$, $E_1 \vdash \gamma(d)$, and $E_2 = E_1[x \leftarrow 0]_{x \in \rho(d) \cup \zeta(d)}$,

(P5) $[\![[E_1], X, [E_2]]\!] \rightarrow [\![succ([E_1]), X, [E_2]]\!]$,

(P6) $[\![[E_1], X, [E_2]]\!] \rightarrow (\!|[F_0], \beta(1), [F_1]|\!) \cdots (\!|[F_{i-1}, \beta(i), [F_i]|\!)[\![[F_i], \beta(i+1), [E_2]]\!]$
if $d = (X, \beta) \in \delta$ with $|\beta| = m > 0$, $i \in \{0, \ldots, m-1\}$,
$E_1 \vdash \gamma(d)$, $F_0 = E_1[x \leftarrow 0]_{x \in \rho(d) \cup \zeta(d)}$, and $[F_1], \ldots, [F_i] \in [\mathcal{E}]$,

(P7) $[\![[E_1], X, [E_1]]\!] \rightarrow \lambda$ if $\Pi(X) \models \pi$.

It can be verified that using the productions (P2), (P3) and (P4), every successful derivation (that generates λ) starting from some nonterminal $(\!|[F], X, [F']|\!)$ corresponds to paths in $\mathcal{R}(\mathcal{M}_\mathcal{O}, \varphi)$ that are of the form:

$$\langle X \cdot \beta, [F]\rangle\langle \mu_1 \cdot \beta, [F_1]\rangle \cdots \langle \mu_m \cdot \beta, [F_m]\rangle\langle \beta, [F']\rangle$$

for some $\beta \in \mathcal{V}ar^+$, and where, for every $i \in \{1, \ldots, m\}$, $\mu_i \in \mathcal{V}ar^+$ and $[F_i] \in [\mathcal{E}]$. Indeed, the productions (P2) correspond to transitions in the region graph due to time progress, whereas the productions (P3) correspond to transitions due to applications of derivation rules in δ ; the productions (P4) ensure in addition the continuity of the paths, i.e., the fact that segments generated from successive location variables constitute effectively an existing path in the region graph.

Furthermore, it can also be verified that using the productions (P5), (P6), and (P7), together with the productions (P2), (P3) and (P4), every successful derivation starting from some nonterminal $[\![[F], X, [F']]\!]$ corresponds to paths in the region graph that are of the form:

$$\langle X \cdot \beta, [F]\rangle\langle \mu_1 \cdot \beta, [F_1]\rangle \cdots \langle \mu_m \cdot \beta, [F_m]\rangle\langle \mu_{m+1} \cdot \beta, [F']\rangle$$

for some $\beta \in \mathcal{V}ar^+$, and where, for every $i \in \{1, \ldots, m\}$, $\mu_i \in \mathcal{V}ar^+$ and $[F_i] \in [\mathcal{E}]$, and $\mu_{m+1} \in \mathcal{V}ar^+$ with $\Pi(\mu_{m+1}(1)) \models \pi$.

Then, it can be deduced that, there exists some path in the region graph starting from $\langle \alpha, [E]\rangle$, and reaching some region $\langle \alpha', [E']\rangle$ where $(\pi \wedge \bigwedge_{i=1}^{n} \xi_i)$ is satisfied, if and only if there exists some rank $i \in \{0, \ldots, |\alpha| - 1\}$, and some equivalence classes $[E_0], \ldots, [E_{i+1}]$, such that $E_0 = E$ and $E_{i+1} \in \mathcal{F}$, and

$$(\!|[E_0],\ \alpha(1), [E_1]|\!) \cdots (\!|[E_{i-1}], \alpha(i), [E_i]|\!)[\![[E_i], \alpha(i+1), [E_{i+1}]]\!] \stackrel{+}{\Longrightarrow} \lambda$$

This is exactly what is expressed by the productions concerning the starting symbol (P1). Then, we obtain the following result:

Proposition 6.1 *$s \models \varphi$ iff $L(\mathcal{G}_{(s,\varphi)}) \neq \emptyset$.*

Finally, it is well known that the emptiness problem for context-free languages is decidable (see for instance [Har78]). Then, we obtain the following decidability result:

Theorem 6.1 *The verification problem of PTC invariance formulas on CFTS's is decidable.*

6.2 Single Integration Constraint

We consider now the case where the observer $\mathcal{O}$ contains accumulators and integrators. The consideration of integration constraints can be done only under some restrictions on the CFTS's and the reachability formulas.

We say that a CFTS is *closed* if all its time guards are conjunctions of non-strict constraints, i.e., of the form $x \leq n$ or $x \geq n$. A single integration constraint (SIC) reachability formula is of the form

$$\exists \Diamond (\pi \wedge \bigwedge_{i=1}^{n} \xi_i)$$

where among the ξ_i's, there is at most one integration constraint and all the time constraints are non-strict (of the form $x \leq n$ or $x \geq n$).

We show in this section that the satisfaction relation between closed CFTS's and SIC reachability formulas is decidable. For this, we show first that in this case, we can check the satisfaction relation by considering only the computations where time takes integer values. Then, we reduce the verification problem to the satisfiability problem of integer linear constraints.

Discrete satisfaction relation

Let $\mathcal{M}$ be a CFTS and $\mathcal{O}$ a LHO of $\mathcal{M}$. We introduce a discrete satisfaction relation between states of $\mathcal{M}_\mathcal{O}$ and formulas. This relation is a specialization of $\models$ which takes into account only states where time has an integer value. Let us give the formal definitions.

A state $s = \langle \alpha, E_\mathcal{T}, E_\mathcal{A}, E_\mathcal{I} \rangle \in \mathcal{S}_{\mathcal{M}_\mathcal{O}}$ is called *integer state* if $\forall u \in \mathcal{T}$, $E_\mathcal{T}(u) \in I\!N$ and $\forall u \in \mathcal{I}$, $E_\mathcal{I}(u) \in \mathbb{Z}$. Let $\mathcal{Z}(\mathcal{S}_{\mathcal{M}_\mathcal{O}})$ be the set of integer states of $\mathcal{M}_\mathcal{O}$.

We can modify in the obvious manner the definition of the relation $\rightarrow$ to obtain the definition of the transition relation $\rightarrow_\mathsf{Z} \subseteq \mathcal{Z}(\mathcal{S}_{\mathcal{M}_\mathcal{O}}) \times \mathcal{Z}(\mathcal{S}_{\mathcal{M}_\mathcal{O}})$ corresponding to discrete (integer) progress of time, and consider the relation $\hookrightarrow_\mathsf{Z} = \rightarrow_\mathsf{Z} \cup \triangleright$. Then, we can define accordingly *integer computation sequences.*

Now, we introduce the discrete satisfaction relation $\models_\mathsf{Z}$ between integer states and invariance formulas (and consequently, reachability formulas). For this, it is sufficient to consider that, for every integer state $s \in \mathcal{Z}(\mathcal{S}_{\mathcal{M}_\mathcal{O}})$, $s \models_\mathsf{Z} \forall\Box\phi$ iff $\forall s'.\ s \overset{*}{\hookrightarrow}_\mathsf{Z} s'.\ s' \models_\mathsf{Z} \phi$.

Obviously, for every integer state and formula φ, $s \models_\mathsf{Z} \varphi$ implies $s \models \varphi$. The reverse implication is far less trivial. We prove this implication by using digitization techniques as in [HMP92, KPSY93] (see the full paper [BER94]). Thus, we have

Proposition 6.2 *Let $s \in \mathcal{Z}(\mathcal{S}_{\mathcal{M}_\mathcal{O}})$ and φ a SIC reachability formula. Then, $s \models \varphi$ iff $s \models_\mathsf{Z} \varphi$.*

From Reachability to Satisfiability

We give now the reduction of the verification problem of SIC reachability formulas to the satisfiability problem of integer linear constraints.

Let $s = \langle \alpha, E_\mathcal{T}, E_\mathcal{A}, E_\mathcal{I} \rangle$ be an integer state, and consider a SIC reachability formula

$$\varphi = \exists\Diamond(\pi \wedge (\bigwedge_{i=1}^{n_1} \xi_i) \wedge (\bigwedge_{i=1}^{n_2} \varrho_i) \wedge \vartheta)$$

where the ξ_i's and the ϱ_i's are respectively time and accumulation constraints, and ϑ is an integration constraint. Suppose that we are interested in the problem of deciding whether $s \models \varphi$.

First of all, recall that by Proposition 6.2, this verification can be done by considering only the integer computations starting from s. Let us focus for the moment on time constraints only, and consider the PTC reachability formula

$$\varphi' = \exists\Diamond(\pi \wedge \bigwedge_{i=1}^{n_1} \xi_i).$$

Then, clearly we can use the verification method shown in Section 6.1 to decide whether $s \models \varphi'$. Recall that it consists in solving a reachability problem in an infinite region graph via a reduction to the nonemptiness problem of a context-free language. However, in the present case, we can consider only regions corresponding to integer valuations of timers (by Proposition 6.2 and since φ' is also a SIC reachability formula).

So, let $\mathcal{E}_\mathsf{Z}$ be the set of time valuations E such that, for every $x \in \mathcal{T}$, $E(x) \in I\!N$ and consider the integer successor function $succ_\mathsf{Z}$ defined by $succ_\mathsf{Z}([E]) = [E+1]$, for every $E \in \mathcal{E}_\mathsf{Z}$. Notice that $succ_\mathsf{Z}$ is always defined and that in the graph $([\mathcal{E}_\mathsf{Z}], succ_\mathsf{Z})$, every path reaches eventually some self-looping equivalence class, when the value of each timer

x becomes greater than the maximal constant compared with x in either time guards of $\mathcal{M}$ or time constraints ξ_i of φ'.

Following the same lines as in Section 6.1, we can define straightforwardly the integer region graph $\mathcal{R}_{\mathbb{Z}}(\mathcal{M}_{\mathcal{O}}, \varphi')$ where the regions involve only equivalence classes in $[\mathcal{E}_{\mathbb{Z}}]$ and the edges are defined by means of the successor function $succ_{\mathbb{Z}}$ instead of $succ$.

Then, it is clear that $s \models \varphi'$ if and only if there exists some path in $\mathcal{R}_{\mathbb{Z}}(\mathcal{M}_{\mathcal{O}}, \varphi')$ which starts from $\langle \alpha, [E_{\mathcal{I}}] \rangle$ and reaches some region that satisfies $(\pi \wedge \bigwedge_{i=1}^{n_1} \xi_i)$. In order to solve this reachability problem, we proceed as in Section 6.1 by reducing it to the nonemptiness problem of a context-free language. Indeed, consider the CFG $\mathcal{G}^{\mathbb{Z}}_{(s,\varphi')}$ defined exactly as in Section 6.1 where the function $succ_{\mathbb{Z}}$ is used instead of the function $succ$. It is clear that $s \models \varphi'$ if and only if $L(\mathcal{G}^{\mathbb{Z}}_{(s,\varphi')}) \neq \emptyset$.

Now, let us see how to take into account the accumulation and integration constraints. Consider the following derivation in $\mathcal{G}^{\mathbb{Z}}_{(s,\varphi')}$:

$$\sigma = \text{“}\omega \Longrightarrow_{p_1} \omega_1 \cdots \Longrightarrow_{p_n} \omega_n \Longrightarrow_{p_{n+1}} \lambda\text{”} \tag{1}$$

where ω and the ω_i's are nonempty sequences of nonterminals (in N^+). From the definition of $\mathcal{G}^{\mathbb{Z}}_{(s,\varphi')}$, necessarily ω_n is constituted by one nonterminal of the form $[\![[F], X, [F]]\!]$ for some location variable X such that $\Pi(X) \models \pi$ and some valuation F such that $F \vdash \bigwedge_{i=1}^{n_1} \xi_i$.

Actually, the verification problem $s \models \varphi$ consists in deciding whether the grammar $\mathcal{G}^{\mathbb{Z}}_{(s,\varphi')}$ has a successful derivation (generating λ) σ (as in 1) such that $\omega = S$ (S is the starting symbol), and furthermore this derivation σ must correspond to some computation sequence which validates the accumulation and integration constraints, given the initial valuations $E_{\mathcal{A}}$ and $E_{\mathcal{I}}$.

We show in the sequel that this decision problem is reducible to solving a set of integer linear constraints $\Omega_{(s,\varphi)}$. Let us define the sets of variables that are involved in this set of constaints. First, with every accumulator $u \in \mathcal{A}$ we associate a variable a_u, and for the unique integrator v appearing in ϑ we associate the variable $\imath_v$. These variables stand for the values of the corresponding accumulators and integrator at the end of the computation represented by σ. Moreover, with every production $p \in Prod$ we associate a variable w_p which stands for the number of applications of p in σ.

Before giving the constraints that constitute $\Omega_{(s,\varphi)}$, let us introduce some notations. Let $Prod_\tau$ be the set of productions (P2) and (P5), and $Prod_\delta$ the set of productions (P3), (P4) and (P6) in the definition of $\mathcal{G}^{\mathbb{Z}}_{(s,\varphi')}$. The productions in $Prod_\tau$ correspond to time progress, whereas every production p in $Prod_\delta$ corresponds to the application of some derivation rule we denote by d_p. Also, for every production p, we denote by X_p the location variable X if $lhs(p)$ is of the form $[\![[F], X, [F']]\!]$ or $\langle\!|[F], X, [F']|\!\rangle$.

Now, we say that a sequence of productions $\mathcal{P} \in Prod^*$ is *elementary* if all its productions apply to different nonterminals, i.e., $\forall p \in Prod.\ |\mathcal{P}|_p \leq 1$. Given a nonterminal A and a sequence $\omega \in N^+$, we define Π^A_ω to be the set of elementary sequences $\mathcal{P}$ on $Prod$ such that $\exists \nu \in N^*, \omega \stackrel{*}{\Longrightarrow}_{\mathcal{P}} A\nu$. Notice that the set Π^A_ω is finite.

We are able now to define $\Omega_{(s,\varphi)}$. First of all, let us define the a_u's and $\imath_v$ in terms of the w_p's. For every accumulator u, we define the linear constraint ACC_u:

$$(0 \leq a_u) \ \wedge \ (a_u = E_{\mathcal{A}}(u) + \sum_{p \in Prod_\delta} \kappa(d_p)(u) \cdot w_p)$$

and concerning the integrator v, we define the constraint INT_v:

$$(0 \leq \imath_v) \wedge (\imath_v = E_{\mathcal{I}}(v) + \sum_{p \in Prod_r} \partial(X_p)(v) \cdot w_p)$$

Now, let us define the constraints on the w_p's. These constraints must be satisfiable by some valuation W of the w_p's, if and only if W corresponds to numbers of applications of productions p's in some existing derivation σ (1) of the grammar.

So, first of all, the constraints on the w_p's must express the fact that any occurrence of a nonterminal appearing along σ must be reduced so that we get the empty sequence λ. Thus, in the derivation σ, for any nonterminal A, the number of the A-reductions, i.e., applications of some productions p such that $lhs(p) = A$ (A-productions), must be equal to the number of the A-introductions, i.e., the number of the occurrences of A in ω and in the right-hand sides of the applied productions. Given a nonterminal A and a sequence $\omega \in N^+$, this fact is expressed by the linear constraint REDUCT^A_ω defined by:

$$\sum_{p \in Prod} |lhs(p)|_A \cdot w_p = |\omega|_A + \sum_{p \in Prod} |rhs(p)|_A \cdot w_p$$

However, some solutions of $\bigwedge_{A \in N} \mathsf{REDUCT}^A_\omega$ may assign to some variable w_p a value which is non null while p is not necessarily involved in some derivation σ.

Indeed, consider some valuation W that validates $\bigwedge_{A \in N} \mathsf{REDUCT}^A_\omega$ and suppose that it corresponds to some derivation σ as in (1). Consider also some nonterminal B which does not appear in ω neither in any ω_i in the considered derivation σ. Now, assume that there is some production $p = B \rightarrow B$ in $Prod$. We can define another valuation W' which assigns to w_p any strictly positive integer and coincides with W on the other variables. Clearly, the new valuation W' is also a solution of $\bigwedge_{A \in N} \mathsf{REDUCT}^A_\omega$. This, solution must be discarded since the values of the accumulators u's and the integrator v, which are calculated from W' using ACC_u and INT_v, do not correspond necessarily to values that can be obtained from some existing derivation of the grammar $\mathcal{G}^{\mathbb{Z}}_{(s,\varphi')}$.

Thus, we must express in addition, the fact that for any nonterminal A, there exists some A-production p with $w_p > 0$ if and only if A appears in ω or in the ω_i's. This is done by the constraint REACH^A_ω:

$$\sum_{p \in Prod} |lhs(p)|_A \cdot w_p > 0 \Leftrightarrow \bigvee_{\mathcal{P} \in \Pi^A_{\mathcal{J}}} \bigwedge_{p \in \mathcal{P}} w_p > 0.$$

Now, the linear system $\Omega_{(s,\varphi)}$ is defined by:

$$\Omega_{(s,\varphi)} = (\bigwedge_{A \in Prod}(\mathsf{REDUCT}^A_S \wedge \mathsf{REACH}^A_S)) \wedge (\bigwedge_{u \in \mathcal{A}} \mathsf{ACC}_u) \wedge \mathsf{INT}_v \wedge (\bigwedge_{i=1}^{n_2} \varrho_i[a_u/u]_{u \in \mathcal{A}}) \wedge \vartheta[\imath_v/v]$$

where, for any constraint ξ, $\xi[y/x]$ denotes the constraint obtained by substituting in ξ each occurrence of x by y. Then, we can prove in a similar way as in [BER93] that:

Proposition 6.3 *$s \models \varphi$ iff $\Omega_{(s,\varphi)}$ is satisfiable.*

Since the satisfiability of integer linear constraints is decidable, we obtain the following decidability result:

Theorem 6.2 *The verification problem of SIC reachability problem on closed CFTS's is decidable.*

7 Conclusion

We propose in this paper automatic verification methods of general invariance properties for infinite timed systems with a context-free underlying structure.

Our specification approach is expressively powerful. It consists in expressing invariance properties by means of observation variables (timers, accumulators and integrators) that are maintained during the execution of the system, i.e., the observation variables can be seen as additional variables that can be modified but never tested by the system.

We have obtained strong decidability results for the verification of such invariance properties on CFTS's. Indeed, when we consider time and accumulation constraints, we have shown that the verification problem of invariance properties on CFTS's is decidable without any restriction. This result generalizes (concerning invariance properties) the one presented in [ACD90] to the case of context-free systems, and the one presented in [BER93] to the case of timed systems. When integration constraints are taken into account, we have shown that the verification problem of *single integration constraints* is decidable for closed CFTS's. In the full paper [BER94], we also show that for CFTS's with one timer that is reset at each derivation, the verification problem of invariance properties is decidable without any restriction on the number of integration constraints. The results concerning integration constraints generalize those given in [KPSY93].

Several extensions of the work presented in this paper can be considered. Among them, it would be interesting to consider the verification problem for other classes of properties like eventuality properties, and more generally, properties expressible in some temporal logic allowing duration as well as occurrence constraints, obtained for instance as a combination of the logics DTL (Duration Temporal Logic) [BES93] and PCTL (Presburger CTL) [BER93].

References

[ACD90] R. Alur, C. Courcoubetis, and D. Dill. Model-Checking for Real-Time Systems. In *LICS'90*. IEEE, 1990.

[ACH93] R. Alur, C. Courcoubetis, and T. A. Henzinger. Computing Accumulated Delays in Real-time Systems. In *Hybrid Systems*, 1993. LNCS 736.

[ACHH93] R. Alur, C. Courcoubetis, T. Henzinger, and P-H. Ho. Hybrid Automata: An Algorithmic Approach to the Specification and Verification of Hybrid Systems. In *Hybrid Systems*, 1993. LNCS 736.

[BBK87] J.C.M. Baeten, J.A. Bergstra, and J.W. Klop. Decidability of Bisimulation Equivalence for Processes Generating Context-Free Languages. Tech. Rep. CS-R8632, 1987. CWI.

[BER93] A. Bouajjani, R. Echahed, and R. Robbana. Verification of Nonregular Temporal Properties for Context-Free Processes. submitted for publication, 1993.

[BER94] A. Bouajjani, R. Echahed, and R. Robbana. Verification of Context-Free Timed Systems using Linear Hybrid Observers. Tech. Rep. Spectre-94-4, Verimag, Grenoble, January 1994.

[BES93] A. Bouajjani, R. Echahed, and J. Sifakis. On Model Checking for Real-Time Properties with Durations. In *LICS'93*. IEEE, 1993.

[BS92] O. Burkart and B. Steffen. Model Checking for Context-Free Processes. In *CONCUR'92*, 1992. LNCS 630.

[Cer92] K. Cerans. Decidability of Bisimulation Equivalence for Parallel Timer Processes. In *CAV'92*, 1992. LNCS 663.

[CES83] E.M. Clarke, E.A. Emerson, and E. Sistla. Automatic Verification of Finite State Concurrent Systems using Temporal Logic Specifications: A Practical Approach. In *POPL'83*. ACM, 1983.

[CGL93] K. Cerans, J. Godskesen, and K. Larsen. Timed Modal Specification: Theory and Tools. In *CAV'93*. LNCS 697, 1993.

[CHR91] Z. Chaochen, C.A.R. Hoare, and A.P. Ravn. A Calculus of Durations. *Information Processing Letters*, 40:269–276, 1991.

[CHS92] S. Christensen, H. Hüttel, and C. Stirling. Bisimulation Equivalence is Decidable for all Context-Free Processes. In *CONCUR'92*, 1992. LNCS 630.

[CS91] R. Cleaveland and B. Steffen. A Linear-Time Model-Checking Algorithm for the Alternation-Free Modal Mu-Calculus. In *Proc. Computer-Aided Verification (CAV'91)*, 1991. LNCS 575.

[EL86] E.A. Emerson and C.L. Lei. Efficient Model-Checking in Fragments of the Propositional μ-Calculus. In *LICS'86*, 1986.

[GH91] J.F. Groote and H. Hüttel. Undecidable Equivalences for Basic Process Algebra. Tech. Rep. ECS-LFCS-91-169, 1991. Dep. of Computer Science, Univ. of Edinburgh.

[Har78] M.A. Harrison. *Introduction to Formal Language Theory*. Addison-Wesley Pub. Comp., 1978.

[HMP92] T. Henzinger, Z. Manna, and A. Pnueli. What Good are Digital Clocks? In *ICALP'92*, 1992. LNCS 623.

[HNSY92] T.A. Henzinger, X. Nicollin, J. Sifakis, and S. Yovine. Symbolic Model-Checking for Real-Time Systems. In *LICS'92*. IEEE, 1992.

[KPSY93] Y. Kesten, A. Pnueli, J. Sifakis, and S. Yovine. Integration Graphs: A Class of Decidable Hybrid System s. In *Hybrid Systems*, 1993. LNCS 736.

[KS83] P. Kanellakis and S.A. Smolka. CCS Expressions, Finite State Processes, and Three Problems of Equivalence. In *PODC'83*. ACM, 1983.

[Mil80] R. Milner. A Calculus of Communication Systems. 1980. LNCS 92.

[MMP92] O. Maler, Z. Manna, and A. Pnueli. A Formal Approach to Hybrid Systems. In *REX workshop on Real-Time: Theory and Practice*, 1992. LNCS 600.

[MP93] Z. Manna and A. Pnueli. Verifying Hybrid Systems. In *Hybrid Systems*, 1993. LNCS 736.

[NOSY93] X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine. An Approach to the Description and Analysis of Hybrid Systems. In *Hybrid Systems*, 1993. LNCS 736.

[NRSV90] X. Nicollin, J.-L. Richier, J. Sifakis, and J. Voiron. ATP: an Algebra for Timed Processes. In *IFIP TC2 Working Conf. on Prog. Concepts and Methods*, 1990. Israel.

[Par81] D. Park. Concurrency and Automata on Infinite Sequences. In *5th GI-Conference on Theoretical Computer Science*. 1981. LNCS 104.

[QS82] J-P. Queille and J. Sifakis. Specification and Verification of Concurrent Systems in CESAR. In *Intern. Symp. on Programming, LNCS 137*, 1982.

[VW86] M.Y. Vardi and P. Wolper. An Automata-Theoretic Approach to Automatic Program Verification. In *LICS'86*. IEEE, 1986.

[Wan90] Y. Wang. Real Time Behaviour of Asynchronous Agents. In *CONCUR'90*, 1990. LNCS 458.