Verifying ω-Regular Properties for a Subclass of Linear Hybrid Systems

Ahmed Bouajjani Riadh Robbana

VERIMAG, Miniparc-Zirst, Rue Lavoisier 38330 Montbonnot St-Martin, France.
Ahmed.Bouajjani@imag.fr, Riadh.Robbana@imag.fr

Abstract. We address the problem of verifying untimed ω-regular properties for a subclass of linear hybrid systems, i.e., finite transition graphs supplied with real-valued variables that change continuously with integer rates at each control location. The systems we consider are systems with two variables, one of them must be monotonic (e.g., with rates either 0 or 1) whereas the other one can have rates either -1, 0, or 1. We prove that for these systems, the verification problem of ω-regular properties is decidable. For that, we show that these systems generate ω-context-free sets of state sequences.

1 Introduction

Hybrid systems are obtained as combinations of discrete and continuous systems. They appear for instance in all the applications where some physical process is controled by a computer.

Natural models of hybrid systems are automata supplied with real-valued variables that change *continuously* at each control location [12, 2, 14]. The *discrete* moves between control locations are conditioned by constraints on the variables of the system, and their execution may reset some of the variables. Linear hybrid systems correspond to the case where all the variables evolve with integer rates. Interesting particular cases of linear hybrid systems have been identified in the literature by imposing some conditions on their discrete and/or continuous dynamics. We mention for instance the *timed graphs* [1] where all the variables are clocks, i.e., their rates are always 1, and *integrator graphs* [2, 11] where all the variables (integrators) may have rates either 0 or 1; examples of linear hybrid systems with restrictions on their discrete dynamics are the *piecewise constant derivative systems* [3]. Timed graphs are widely used for the description of real-time systems. They allow to constrain the time distance between two events. Integrator graphs allow to reason about the more general notion of *duration* of state properties [4]. The duration of a property in some computation segment is the accumulated time that this property holds in the computation.

In this paper, we address the verification problem of *extended integrator graphs* (EIG's), where the variables may have rates either -1, 0, or 1 ($\{-1, 0, 1\}$-variables). The consideration of negative rates allows for instance to reason about differences between durations.

The existing results on the verification of subclasses of linear hybrid systems consider in their majority the case of invariance properties. The verification problem for these properties reduces to (the complement of) the reachability

problem in the considered classes of systems. It has been shown that this problem is undecidable even for integrator graphs with only one integrator (and many clocks) [10]. As for EIG's, it suffices to consider the subclass of systems with one clock and two $\{-1, 0, 1\}$-variables to simulate any 2-counter machine [11].

In this work, we consider the case of two dimentional EIG's (2-dim EIG's), i.e., EIG's with two variables. We prove mainly that for 2-dim EIG's with one integrator and one $\{-1, 0, 1\}$-variable (single-integrator 2-dim EIG's), the verification problem of all untimed ω-regular (Muller ω-automata definable) properties is decidable. We obtain this result by proving that the sets of state sequences (untimed traces) generated by these systems are ω-context-free (pushdown ω-automata definable). Then, we use the fact that the inclusion problem of ω-context-free languages in ω-regular ones is decidable [6].

To prove that single-integrator 2-dim EIG's generates ω-context-free sets of state sequences, we adopt a partition-based technique. We define an *infinite* partition of the dense space of valuations of the variables ($\mathbb{R}^2$) which is *boundedly finite*, i.e., *finite* on every *bounded* subplan of $\mathbb{R}^2$, and we prove that starting from every control location with two valuations in the same class, the same sets of state sequences can be generated. In other words, the partition we consider is compatible with trace equivalence, and despite the fact that it is infinite, it is encodable using one counter. However, we show that this partition is not a bisimulation and cannot be used to reason about branching-time properties in general. Moreover, we prove that there exist single-integrator 2-dim EIG's such that there is no boundedly finite bisimulation on their transition graph.

The remainder of this paper is organized as follows. In Section 2, we introduce the EIG's. In Section 3, we recall the definition of ω-automata. In Section 4, we introduce the logic ECTL* which is used to express linear-time as well as branching-time untimed properties of EIG's. In Section 5, we show the relation between behavioural equivalences as trace equivalence and bisimulation, and logical equivalences corresponding to fragments of ECTL*. In Section 6, we introduce some basic partitions of the 2-dim valuations ($\mathbb{R}^2$). Sections 7 and 8 concern the existence of property preserving boundedly finite partitions. In Section 7, we consider the simple case of 2-dim integrator graphs, and in Section 8, we investigate the more general case of single-integrator 2-dim EIG's. Section 9 is dedicated to the verification problem. Finally, concluding remarks are given in Section 10.

2 Extended Integrator Graphs

We introduce in this section models of hybrid systems, we call *extended integrator graphs* (EIG for short), that are particular cases of *linear hybrid systems* introduced in [12, 2, 14]. Roughly speaking, they consist of control transition graphs with finitely many locations, supplied with real valued variables that change continuously at each control location with rates in $\{-1, 0, 1\}$. These models generalize the so-called *timed graphs* [1] where all the variables change with rate 1, and *integrator graphs* where the variables may have rates in $\{0, 1\}$ [2, 11].

Before giving the formal definition of EIG's, let us introduce *simple linear constraints* that are used in their enabling guards. Let $\mathcal{V}$ be a set of real valued

variables. A *simple linear constraint* over $\mathcal{V}$ is a boolean combination of constraints of the form $x \sim c$ where $x \in \mathcal{V}$, c is an integer constant ($c \in \mathbb{Z}$), and $\sim \in \{<, \leq\}$; the symbols $<$ and $\leq$ represent the usual (strict and nonstrict) ordering relations over reals. Let $\mathcal{C}onst(\mathcal{V})$ be the set of simple linear constraints over $\mathcal{V}$. We use letters $f, g, \cdots$ to range over the set $\mathcal{C}onst(\mathcal{V})$. A *valuation* over $\mathcal{V}$ is a function in $[\mathcal{V} \rightarrow \mathbb{R}]$. A satisfaction relation is defined as usual between valuations and constraints. Given valuation ν and $f \in \mathcal{C}onst(\mathcal{V})$, we denote by $\nu \models f$ the fact that ν satisfies f, and for every $X \subseteq \mathcal{V}$, we denote by $\nu[X \mapsto 0]$ the valuation which associates with each variable in X the value 0, and coincides with ν on all the other variables.

Now, let $\mathcal{P}$ be a finite set of atomic propositions and $\Sigma = 2^{\mathcal{P}}$. We call *state* any element of Σ, and *state sequence* any infinite sequence in Σ^{ω}.

An extended integrator graph $\mathcal{H}$ over Σ consists of the following components:

- $\mathcal{X}$, a finite set of variables,
- $\mathcal{L}$, a finite set of control locations,
- $\mathcal{E}$, a set of edges. Each edge is a tuple (ℓ, g, X, ℓ') where $\ell, \ell' \in \mathcal{L}$ are the source and target locations, $g \in \mathcal{C}onst(\mathcal{X})$ is an enabling guard, and $X \subseteq \mathcal{X}$ is the set of the reset variables,
- Π, a function in $[\mathcal{L} \rightarrow \Sigma]$, associating a set of atomic propositions (a state) with each location,
- ∂, a function in $[\mathcal{L} \times \mathcal{X} \rightarrow \{-1, 0, 1\}]$, associating with each location ℓ and variable x, a rate at which x changes continuously at ℓ.

We say that a variable x is a *clock* (resp. *integrator*) if for every control location ℓ, we have $\partial(\ell, x) = 1$ (resp. $\partial(\ell, x) \in \{0, 1\}$). Then, a *timed graph* (TG) is an EIG such that all its variables are clocks, whereas an *integrator graph* (IG) is an EIG such that all its variables are integrators. Finally, for every $n \geq 1$, an n-dim EIG (resp. n-dim IG) is an EIG (resp. IG) with exactly n variables.

We define hereafter the notions of configuration, computation sequence, and state sequence of an EIG.

A *configuration* of $\mathcal{H}$ is a pair $\langle \ell, \nu \rangle$ where $\ell \in \mathcal{L}$, and $\nu \in [\mathcal{X} \rightarrow \mathbb{R}]$ is a valuation over $\mathcal{X}$. We denote by $\mathcal{C}onf(\mathcal{H})$ the set of all possible configurations of $\mathcal{H}$. A *timed configuration* of $\mathcal{H}$ is a triplet $\langle \ell, \nu, t \rangle$ where $\langle \ell, \nu \rangle$ is a configuration, and $t \in \mathbb{R}_{\geq 0}$ is a time stamp.

Given a valuation ν of the variables, a control location ℓ, and $t \in \mathbb{R}_{\geq 0}$, we denote by $[\nu + t]_{\ell}$ the valuation ν' such that $\forall x \in \mathcal{X}.\ \nu'(x) = \nu(x) + \partial(\ell, x) \cdot t$, i.e., the valuation of the variables obtained from ν by staying at location ℓ for an amount of time t. Then, a *timed computation sequence* of $\mathcal{H}$ starting from a configuration $\langle \ell, \nu \rangle$ is an infinite sequence of timed configurations $(\langle \ell_i, \nu_i, t_i \rangle)_{i \in \omega}$ such that $\langle \ell, \nu, 0 \rangle = \langle \ell_0, \nu_0, t_0 \rangle$, $\forall i \in \omega.\ t_{i+1} \geq t_i$, $\lim_{i \rightarrow \infty} t_i = \infty$, and $\forall i \in \omega$,

- either $\ell_i = \ell_{i+1}$ and $\nu_{i+1} = [\nu_i + (t_{i+1} - t_i)]_{\ell_i}$,
- or $t_i = t_{i+1}$ and $\exists (\ell_i, g, X, \ell_{i+1}) \in \mathcal{E}.\ \nu_i \models g$, and $\nu_{i+1} = \nu_i[X \mapsto 0]$.

Notice that the condition $\lim_{i \rightarrow \infty} t_i = \infty$ ensures that time can go beyong any value; timed computation sequences are then called *time diverging*.

Every timed computation sequence generates a *computation sequence* by abstracting from time stamps: $(\langle \ell_i, \nu_i, t_i \rangle)_{i \in \omega}$ generates $(\langle \ell_i, \nu_i \rangle)_{i \in \omega}$. Then, given a configuration $\kappa \in \mathcal{C}onf(\mathcal{H})$, we denote by $\mathcal{C}omp(\kappa, \mathcal{H})$ the set of computation sequences generated by the timed computations of $\mathcal{H}$ starting from κ.

Finally, every computation sequence $(\langle \ell_i, \nu_i \rangle)_{i \in \omega}$ generates a state sequence $(\Pi(\ell_i))_{i \in \omega}$. Given a configuration κ, We denote by $\mathcal{T}(\kappa, \mathcal{H})$ the set of state sequences generated by the computation sequences in $\mathcal{C}omp(\kappa, \mathcal{H})$.

3 Muller ω-automata

We recall in this section the definition of Muller ω-automata [13] and some basic results concerning them which are relevant to this paper.

Let V be a finite alphabet. A Muller ω-automaton over V is a tuple $\mathcal{A} = (\mathcal{Q}, q_I, \delta, \mathcal{F})$ where $\mathcal{Q}$ is a finite set of control states, $q_I \in \mathcal{Q}$ is the initial control state, $\delta \subseteq \mathcal{Q} \times V \times \mathcal{Q}$ is a labelled transition relation, and $\mathcal{F} \subseteq 2^{\mathcal{Q}}$ is a collection of *repetition sets*. Given a sequence $\sigma = (s_i)_{i \in \omega} \in V^\omega$, a run of $\mathcal{A}$ over σ is an infinite sequence $\rho = (q_i)_{i \in \omega}$ such that $q_0 = q_I$, and $\forall i \in \omega.\ (q_i, s_i, q_{i+1}) \in \delta$. Moreover, the run ρ over σ is *accepting* if $\{q \in \mathcal{Q} : \overset{\infty}{\exists}\, i \in \omega.\ q_i = q\} \in \mathcal{F}$. Then, a sequence $\sigma \in V^\omega$ is accepted by the automaton $\mathcal{A}$ if it has an accepting run in $\mathcal{A}$. We denote by $L(\mathcal{A})$ the set of sequences accepted by $\mathcal{A}$. Muller ω-automata definable sets are called *ω-regular languages*. It is well known that the class of ω-regular languages is closed under all boolean operations, and that the emptiness problem of ω-regular languages is decidable [17].

The definition above can be extended staightforwardly (as for automata on finite words) to 1-counter Muller ω-automata and Pushdown Muller ω-automata, the formers being obviously particular cases of the latters. Pushdown Muller ω-automata define *ω-context-free languages*. It has been shown that the problem whether some ω-context-free language is included in some ω-regular one is decidable; this is due to the fact that the intersection of an ω-context-free language with an ω-regular language is an ω-context-free language, and that the emptiness problem of ω-context-free languages is decidable [6].

4 The logic ECTL*

We introduce in this section the branching-time extended temporal logic ECTL* [7, 16]. This logic is defined as the union of stratified logics ECTL^*_n for every $n \in \omega$. The set of formulas of ECTL^*_0 is simply the set of atomic propositions $\mathcal{P}$, and for every $n \geq 1$, the set of formulas of ECTL^*_n is the smallest set of formulas such that:

- $\mathrm{ECTL}^*_{n-1} \subseteq \mathrm{ECTL}^*_n$,
- if $\varphi \in \mathrm{ECTL}^*_n$, then $\neg\varphi \in \mathrm{ECTL}^*_n$,
- if $\varphi_1, \varphi_2 \in \mathrm{ECTL}^*_n$, then $\varphi_1 \vee \varphi_2 \in \mathrm{ECTL}^*_n$,
- if Φ is a finite set of ECTL^*_{n-1} formulas, and $\Omega(\Phi)$ is an ω-regular language over the alphabet 2^Φ, then $\exists\Omega(\Phi) \in \mathrm{ECTL}^*_n$.

We consider as abbreviations the usual boolean connectives as conjunction ($\wedge$) and implication ($\Rightarrow$), as well as $\forall\Omega(\Phi) = \neg\exists\overline{\Omega}(\Phi)$ where $\overline{\Omega}(\Phi) = 2^\Phi - \Omega(\Phi)$.

Let $\mathcal{H}$ be an EIG. Then, formulas of ECTL* are interpreted as sets of configurations of $\mathcal{H}$. Intuitively, the formulas $\exists\Omega(\Phi)$ represents the set of configurations from which there exists a computation sequence $(\kappa_i)_{i\in\omega}$ satisfying the *path constraint* $\Omega(\Phi)$, i.e., the sequence in $(2^\Phi)^\omega$ corresponding to the subsets of Φ that are satisfied successively by the κ_i's belongs to the language $\Omega(\Phi)$. The interpretation of ECTL* formulas is defined inductively in the following manner:

- $[\![P]\!] = \{\langle \ell, \nu\rangle \in \mathcal{C}onf(\mathcal{H}) \ : \ P \in \Pi(\ell)\}$,
- $[\![\neg\varphi]\!] = \{\kappa \in \mathcal{C}onf(\mathcal{H}) \ : \ \kappa \notin [\![\varphi]\!]\}$,
- $[\![\varphi_1 \vee \varphi_2]\!] = [\![\varphi_1]\!] \cup [\![\varphi_2]\!]$,
- $[\![\exists\Omega(\Phi)]\!] = \{\kappa \in \mathcal{C}onf(\mathcal{H}) \ : \ \exists(\kappa_i)_{i\in\omega} \in \mathcal{C}omp(\kappa, \mathcal{H}).\ (T_\Phi(\kappa_i))_{i\in\omega} \in \Omega(\Phi)\}$, where, for every $i \in \omega$, $T_\Phi(\kappa_i) = \{\varphi \in \Phi \ : \ \kappa_i \in [\![\varphi]\!]\}$.

It is easy to see that the branching-time temporal logics CTL [5] and CTL* [8] are less expressive than ECTL*. For every $n \in \omega$, let CTL_n and CTL_n^* be the fragments of ECTL_n^* corresponding respectively to CTL and CTL* formulas with at most n nested path quantifiers ($\exists$'s).

It can be observed also that ECTL* can express all linear-time ω-regular properties. Indeed, these properties correspond to ECTL_1^* formulas of the form $\forall\Omega(\mathcal{P})$. Hence, ECTL_1^* subsumes linear-time temporal logics as PTL [15] and ETL [19], as well as the linear-time μ-calculus [18].

5 Behavioural vs. Logical Equivalences

We present in this section results relating bisimulation-based behavioural equivalences with the equivalences induced by the logic ECTL* and its fragments ECTL_n^*, for every $n \in \omega$.

First of all, we introduce the notion of *transition graph* associated with an EIG. Let us fix an EIG $\mathcal{H}$. We consider the transition relation between the configurations of $\mathcal{H}$ defined by: $\langle \ell, \nu\rangle \rhd \langle \ell', \nu'\rangle$ iff either $\ell = \ell'$ and $\exists t \in \mathbb{R}_{\geq 0}.\ \nu' = [\nu + t]_\ell$, or $\exists(\ell, g, X, \ell') \in \mathcal{E}.\ \nu \models g$ and $\nu' = \nu[X \mapsto 0]$. Then, the transition graph associated with $\mathcal{H}$ is $\mathcal{G}_\mathcal{H} = (\mathcal{C}onf(\mathcal{H}), \rhd)$.

Intuitively, the relation $\rhd$ represents either time progress transitions at a same location or discrete transitions between control locations. Notice that for every computation sequence $(\langle \ell_i, \nu_i\rangle)_{i\in\omega}$, $\forall i \in \omega.\ \langle \ell_i, \nu_i\rangle \rhd \langle \ell_{i+1}, \nu_{i+1}\rangle$.

A *bisimulation* over $\mathcal{G}_\mathcal{H}$ is any symetric binary relation R between configurations of $\mathcal{H}$ such that $\langle \ell_1, \nu_1\rangle R \langle \ell_2, \nu_2\rangle$ iff

- $\Pi(\ell_1) = \Pi(\ell_2)$, and
- $\langle \ell_1, \nu_1\rangle \rhd \kappa_1$ implies that $\exists\kappa_2.\ \langle \ell_2, \nu_2\rangle \rhd \kappa_2$ and $\kappa_1 R \kappa_2$.

We denote by $\sim$ the greatest bisimulation over $\mathcal{G}_\mathcal{H}$. It can be easily verified that $\sim$ is an equivalence. The relation $\sim$ can be extended pointwisely to computation sequences. The following lemma relates bisimilarity between configurations and bisimilarity between computation sequences.

Lemma 5.1 *$\forall\kappa, \kappa' \in \mathcal{C}onf(\mathcal{H})$, $\kappa \sim \kappa'$ implies that $\forall(\kappa_i)_{i\in\omega} \in \mathcal{C}omp(\kappa, \mathcal{H})$, $\exists(\kappa'_i)_{i\in\omega} \in \mathcal{C}omp(\kappa', \mathcal{H})$ such that $\forall i \in \omega.\ \kappa_i \sim \kappa'_i$.*

Using Lemma 5.1, the following preservation result can be proved by induction on the structure of ECTL* formulas.

Proposition 5.1 *For every configurations κ and κ', and every ECTL* formula φ, $\kappa \sim \kappa'$ implies that $\kappa \in [\![\varphi]\!]$ iff $\kappa' \in [\![\varphi]\!]$.*

Now, we introduce a decreasing family $(\sim_n)_{n\in\omega}$ of equivalences between configurations, and show that for every $n \in \omega$, $\sim_n$ is compatible with (included in) the equivalence induced by ECTL^*_n. The family $(\sim_n)_{n\in\omega}$ is defined by:

- $\langle \ell, \nu \rangle \sim_0 \langle \ell', \nu' \rangle$ iff $\Pi(\ell) = \Pi(\ell')$,
- $\forall n \in \omega$, $\kappa_1 \sim_{n+1} \kappa_2$ iff
  - $\forall (\kappa_i)_{i\in\omega} \in \mathcal{C}omp(\kappa, \mathcal{H}).\, \exists (\kappa'_i)_{i\in\omega} \in \mathcal{C}omp(\kappa', \mathcal{H}).\, \forall i \in \omega.\, \kappa_i \sim_n \kappa'_i$,
  - $\forall (\kappa'_i)_{i\in\omega} \in \mathcal{C}omp(\kappa', \mathcal{H}).\, \exists (\kappa_i)_{i\in\omega} \in \mathcal{C}omp(\kappa, \mathcal{H}).\, \forall i \in \omega.\, \kappa_i \sim_n \kappa'_i$.

It can be verified that for every $n \in \omega$, $\sim_n$ is indeed an equivalence. Notice that for every configurations κ and κ', $\kappa \sim_1 \kappa'$ iff $\mathcal{T}(\kappa, \mathcal{H}) = \mathcal{T}(\kappa', \mathcal{H})$, i.e., $\sim_1$ corresponds to trace equivalence.

We can prove similarly to Proposition 5.1, using induction on natural numbers, that for every $n \in \omega$, the equivalence $\sim_n$ preserves the formulas of ECTL^*_n.

Proposition 5.2 *For every $n \in \omega$, every configurations κ and κ', and every ECTL^*_n formula φ, $\kappa \sim_n \kappa'$ implies that $\kappa \in [\![\varphi]\!]$ iff $\kappa' \in [\![\varphi]\!]$.*

6 Basic Partitions

We focus in the sequel on the case of 2-dim EIG's. Let $\mathcal{H}$ be such a system, and let x and y be the its two variables.

We introduce here some basic equivalences on $[\{x, y\} \to \mathbb{R}]$ that will be combined in the next sections to define compatible equivalences with fragments of ECTL*. These basic equivalences are parametrized by a natural number n; they correspond to slicing the plan $\mathbb{R}^2$ w.r.t. the unit $1/2^n$, either according to horizontal or to vertical lines, or according to positive or to negative diagonals.

Let us first define the horizontal slicing. For every $n \in \omega$, we define an equivalence $\rightarrow_n$ between valuations in the following manner: $\forall \nu, \nu' \in [\{x, y\} \to \mathbb{R}]$, $\nu \rightarrow_n \nu'$ if $\forall k \in \mathbb{Z}$,

- $\nu(y) = k/2^n$ iff $\nu'(y) = k/2^n$, and
- $k/2^n < \nu(y) < (k+1)/2^n$ iff $k/2^n < \nu'(y) < (k+1)/2^n$.

The vertical slicing is defined in a similar way. For that, we introduce, for every $n \in \omega$, an equivalence between valuations denoted by $\uparrow_n$ and which is defined as the relation $\rightarrow_n$ above by considering the variable x instead of y.

Now, let us define the slicing according to positive diagonals. It corresponds to an equivalence $\nearrow_n$ which is defined, for every $n \in \omega$, in the following manner: $\forall \nu, \nu' \in [\{x, y\} \to \mathbb{R}]$, $\nu \nearrow_n \nu'$ if $\forall k \in \mathbb{Z}$,

- $\nu(y) - \nu(x) = k/2^n$ iff $\nu'(y) - \nu'(x) = k/2^n$, and

– $k/2^n < \nu(y) - \nu(x) < (k+1)/2^n$ iff $k/2^n < \nu'(y) - \nu'(x) < (k+1)/2^n$.

The slicing according to negative diagonals corresponds to an equivalence $\searrow_n$ which is defined, for every $n \in \omega$, as the relation $\nearrow_n$ above by replacing "$\nu(y) - \nu(x)$" (resp. "$\nu'(y) - \nu'(x)$") by "$\nu(y) + \nu(x)$" (resp. "$\nu'(y) + \nu'(x)$").

We define other basic equivalences obtained from the ones above by introducing a constraint which determines the subplan where the slicing is applied. Let $\rightsquigarrow$ stands for either $\rightarrow$, $\uparrow$, $\nearrow$, or $\searrow$. Then, given $f \in \mathcal{C}onst(\{x,y\})$, we have $\forall \nu, \nu' \in [\{x,y\} \rightarrow I\!R]$, $\nu \rightsquigarrow_n^f \nu'$ if ($\nu \models f$ iff $\nu' \models f$, and if $\nu \models f$ then $\nu \rightsquigarrow_n \nu'$).

7 A simple case: 2-dim integrator graphs

We start by considering the relatively simple case of 2-dim IG's. We show that there exists a finite-index equivalence $\approx$ on valuations which induces on the configurations of any 2-dim IG an equivalence that preserves all the formulas of ECTL*, i.e., such that configurations with the same location and $\approx$-equivalent valuations satisfy the same ECTL* formulas.

Let $\mathcal{H}$ be a 2-dim IG, and let x and y be its integrators. First of all, let us introduce some notations. Let z stands for x or y. Then, we denote by c_z the greatest constant which is compared with z in the guards of $\mathcal{H}$, and we denote by f_z the constraint $0 \leq z \leq c_z$.

Then, the equivalence $\approx$ is defined by $\approx = -\!\!\!\rightarrow_0^{f_y} \cap \uparrow_0^{f_x} \cap \nearrow_0^{f_x \wedge f_y}$. Notice that $\approx$ is actually the same equivalence considered in [1] to reason about timed graphs.

It can be seen that for every valuations ν and ν', and every guard g, if $\nu \approx \nu'$ then $\nu \models g$ iff $\nu' \models g$. This is due to the fact that variables are compared only with integer constants, and that since the variables x and y are monotonic (never decrease), all their values beyond the constants c_x and c_y respectively satisfy the same guards. Then, it can be proved straightforwardly that $\approx$ induces a bisimulation over the transition graph $\mathcal{G}_{\mathcal{H}}$.

Lemma 7.1 *For every locations ℓ and ℓ', and every valuations ν, ν' and μ, $\nu \approx \nu'$ implies that if $\langle \ell, \nu \rangle \rhd \langle \ell', \mu \rangle$ then there exists a valuation μ' such that $\langle \ell, \nu' \rangle \rhd \langle \ell', \mu' \rangle$ and $\mu \approx \mu'$.*

Since $\sim$ is the greatest bisimulation over $\mathcal{G}_{\mathcal{H}}$, we deduce from Lemma 7.1 that if $\nu \approx \nu'$, then $\langle \ell, \nu \rangle \sim \langle \ell, \nu' \rangle$ for every location ℓ. Then, by Proposition 5.1, we obtain the following result.

Proposition 7.1 *For every location ℓ of $\mathcal{H}$, every valuations ν and ν', and every ECTL* formula φ, $\nu \approx \nu'$ implies that $\langle \ell, \nu \rangle \in [\![\varphi]\!]$ iff $\langle \ell, \nu' \rangle \in [\![\varphi]\!]$.*

8 One integrator and one nonmonotonic variable

We consider now the case of 2-dim EIG's with one integrator and one $\{-1, 0, 1\}$-variable (single-integrator 2-dim EIG's). We prove that in general, there is no

partition of the set of valuations ($\mathbb{R}^2$) that preserves ECTL*, and which is *boundedly finite*, i.e., finite on every bounded subplan of $\mathbb{R}^2$. Actually, we show that this fact holds even if we only consider the fragment CTL_2^*. Indeed, we exhibit a system where CTL_2^* formulas can distinguish configurations with arbitrarily close valuations (points in $\mathbb{R}^2$). On the other hand, we prove that there exists a boundedly finite equivalence on valuations which preserves all the formulas in ECTL_1^*, for any system. This allows to consider all ω-regular properties.

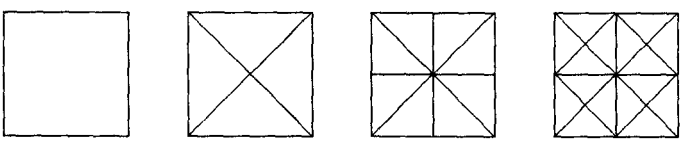
Let us fix for the remainder of this section a 2-dim EIG $\mathcal{H}$ with two variables x and y, and suppose that x is an integrator (y being any $\{-1, 0, 1\}$-variable).

8.1 A family of partitions

We introduce a decreasing family $(\simeq_n)_{n\in\omega}$ of equivalences on valuations defined as follows:

- $\simeq_0 = \mathbb{R}^2$,
- $\forall n \geq 0,\ \simeq_{2n+1} = \simeq_{2n} \cap \rightarrow_n \cap \uparrow_n^{f_x}$,
- $\forall n \geq 1,\ \simeq_{2n} = \simeq_{2n-1} \cap \nearrow_n^{f_x} \cap \searrow_n^{f_x}$.

The following picture represents some members of the family $(\simeq_n)_{n\in\omega}$



$\simeq_1$ $\simeq_2$ $\simeq_3$ $\simeq_4$

Notice that each equivalence $\simeq_n$ is boundedly finite. For every $n \in \omega$, we call $\simeq_n$-*boundary region* any equivalence class of $\simeq_n$ which is either a point or a line. Notice that, $\simeq_1$-boundary regions are equivalence classes of $\simeq_2$. Moreover, observe that $\simeq_1$ is compatible with the equivalence induced by the guards of $\mathcal{H}$, i.e., for every valuations ν and ν', $\nu \simeq_1 \nu'$ implies that for every guard g of $\mathcal{H}$, $\nu \models g$ iff $\nu' \models g$. This is due to the fact that variables are tested with integer constants, and also to the fact that all the values of x beyond c_x satisfy the same guards because x is a monotonic variable. It is clear that we cannot bound the partition $\simeq_1$ in the dimension of y (as we do concerning x) since y is nonmonotonic and unbounded.

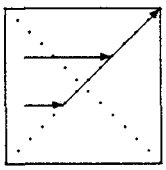
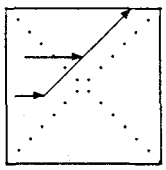
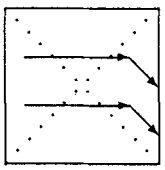
8.2 A compatible partition for ECTL_1^*

We show hereafter that $\simeq_2$ induces on configurations an equivalence which is included in $\sim_1$, and hence, it preseves ECTL_1^*.

First of all, using the fact that $\simeq_1$ is compatible with the equivalence induced by the guards of $\mathcal{H}$, and that $\simeq_1$-boundary regions are equivalence classes of $\simeq_2$, we can prove the following lemma.

Lemma 8.1 *For every valuations ν and ν', if $\nu \simeq_2 \nu'$ then, whenever $\langle \ell, \nu\rangle = \langle \ell_0, \nu_0\rangle \rhd \cdots \langle \ell_m, \nu_m\rangle \rhd \langle \ell', \mu\rangle$ with $\forall i \in \{0, \cdots, m\}.\ \nu \simeq_1 \nu_i$ and μ is in some $\simeq_1$-boundary region, necessarily $\langle \ell, \nu'\rangle = \langle \ell_0, \nu'_0\rangle \rhd \cdots \langle \ell_m, \nu'_m\rangle \rhd \langle \ell', \mu'\rangle$ where $\forall i \in \{0, \cdots, m\}.\ \nu_0 \simeq_1 \nu'_i$, and $\mu \simeq_2 \mu'$.*

The following pictures illustrate Lemma 8.1 on some typical examples.

Notice that Lemma 8.1 does not hold when we consider a target configuration $\langle \ell', \mu \rangle$ where μ is not necessarily in some $\simeq_1$-boundary region.

Now, let $\kappa = \langle \ell, \nu \rangle$ and $\kappa' = \langle \ell, \nu' \rangle$ be two configurations such that $\nu \simeq_2 \nu'$, and consider a computation sequence $\rho = (\langle \ell_i, \nu_i \rangle)_{i \in \omega}$ starting from κ. Let us suppose without loss of generality that ρ is such that for every two configurations $\langle \ell_i, \nu_i \rangle$ and $\langle \ell_j, \nu_j \rangle$ with $j > i$, such that ν_i and ν_j are in two different non $\simeq_1$-boundary regions, there exists some configuration $\langle \ell_k, \nu_k \rangle$ with $i < k < j$, such that ν_k is in some $\simeq_1$-boundary region. By the fact that time increases beyond any integer in a computation sequence (they are genetared by time-divergent timed computation sequences), and since at any time, a variable with rate 1 or -1 will eventually take some integer value (unless it is reset, and then it becomes integer), it can be seen that necessarily, either there exists an infinite sequence of indices $(i_j)_{j \in \omega}$ such that $\forall j \in \omega.\ \nu_{i_j}$ is in some $\simeq_1$-boundary region, or a finite sequence of indices $(i_j)_{j=1}^{m}$, such that $\forall j.\ 0 \leq j \leq m-1.\ \nu_{i_j}$ is in some $\simeq_1$-boundary region, $\nu_{i_m}(x) > c_x$, $fract(\nu_{i_m}(y)) \neq 0$, and $\forall i > i_{m-1}.\ \nu_i \simeq_1 \nu_{i_m}$. Using Lemma 8.1 and the fact that $\simeq_1$-equivalent valuations satisfy the same guards, we can prove that in both cases, there exists a computation sequence of κ' which visits exactly the same locations, i.e., we have the following result.

Lemma 8.2 *For every location ℓ, and every valuations ν and ν', $\nu \simeq_2 \nu'$ implies that $\forall (\langle \ell_i, \nu_i \rangle)_{i \in \omega} \in Comp(\langle \ell, \nu \rangle, \mathcal{H}).\ \exists (\langle \ell_i, \nu'_i \rangle)_{i \in \omega} \in Comp(\langle \ell, \nu' \rangle, \mathcal{H})$.*

Notice that for every valuations ν and ν', and every location ℓ, we have trivially $\langle \ell, \nu \rangle \sim_0 \langle \ell, \nu' \rangle$. Then, using Lemma 8.2, we deduce that for every valuations ν and ν', $\nu \simeq_2 \nu'$ implies that $\langle \ell, \nu \rangle \sim_1 \langle \ell, \nu' \rangle$ for every location ℓ. In other words, $\simeq_2$ is included in trace equivalence, i.e., $\nu \simeq_2 \nu'$ implies that $\mathcal{T}(\langle \ell, \nu \rangle, \mathcal{H}) = \mathcal{T}(\langle \ell, \nu' \rangle, \mathcal{H})$. Hence, by Proposition 5.2, we deduce:

Proposition 8.1 *Let $\mathcal{H}$ be a 2-dim EIG where one of the variables (x) is an integrator. Then, for every location ℓ, every valuations ν and ν', and every $ECTL_1^*$ formula φ, $\nu \simeq_2 \nu'$ implies that $\langle \ell, \nu \rangle \in [\![\varphi]\!]$ iff $\langle \ell, \nu' \rangle \in [\![\varphi]\!]$.*

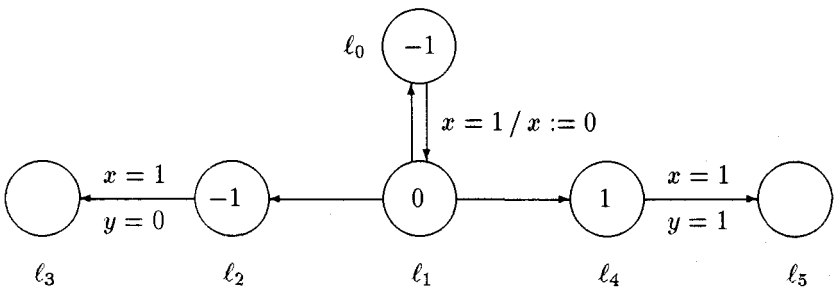
8.3 Uncompatibility with CTL_2^*

It can be observed that, for every $n \in \omega$, there exists a system where the equivalence $\simeq_n$ does not induce a bisimulation on its configurations. For instance, it can be seen on the pictures illustrating Lemma 8.1 that from two $\simeq_2$-equivalent points we can reach different $\simeq_2$-boundary regions through control locations

where y has the rate 0. Hence, each of the equivalences $\simeq_n$ is not suitable for the verification of general branching-time properties.

We prove that actually, there is no boundedly finite partition of the set of valuations which induces a bisimulation on the configurations of every system. Indeed, we show that there exists a system $\mathcal{S}$ and a family of CTL_2^* formulas $(\varphi_n)_{n\geq 2}$ such that, for every $n \geq 2$, there are configurations of $\mathcal{S}$ with $\simeq_n$-equivalent valuations which are distinguished by the formula φ_n.

The system $\mathcal{S}$ is represented in the figure below. In this system, the variable x is a clock, and hence, we omit its rate at each control location (it is always equal to 1) and give only the rates of the variable y (when they are relevant).



Now, let us define the family of formulas $(\varphi_n)_{n\geq 2}$. First, let us introduce the CTL_1 formulas $\phi_0 = \exists((\text{at-}\ell_1 \vee \text{at-}\ell_2)\mathcal{U}\,\text{at-}\ell_3)$, and $\phi_1 = \exists((\text{at-}\ell_1 \vee \text{at-}\ell_4)\mathcal{U}\,\text{at-}\ell_5)$. Then, for every $n \geq 2$, we define the CTL_2^* formula $\varphi_n = \exists\psi_n$, where

- $\psi_2 = \phi_0 \mathcal{U}(\text{at-}\ell_4 \wedge \phi_1)$,
- $\forall k \geq 1.\ \psi_{2k+1} = \text{at-}\ell_0 \mathcal{U}(\text{at-}\ell_1 \wedge \psi_{2k})$,
- $\forall k \geq 2.\ \psi_{2k} = \text{at-}\ell_1 \mathcal{U}(\text{at-}\ell_1 \wedge \neg\phi_1 \wedge (\text{at-}\ell_1 \mathcal{U} \text{at-}\ell_0 \wedge \psi_{2k-1}))$.

Let us see how the formula φ_2 distinguishes between configurations with $\simeq_2$-equivalent valuations. First of all, consider the formulas ϕ_0 and ϕ_1 involved in φ_2. The formula ϕ_0 (resp. ϕ_1) is satisfied by the configurations from which, without visiting ℓ_0, the configuration $\langle \ell_3, (1,0)\rangle$ (resp. $\langle \ell_5, (1,1)\rangle$) is reachable. Then, ϕ_0 (resp. ϕ_1) is satified by precisely the configurations that are either at location ℓ_1 and satisfying $y + x \leq 1$ (resp. $y - x \geq 0$), or at location ℓ_2 (resp. ℓ_4) and satisfying $y + x = 1$ (resp. $y - x = 0$).

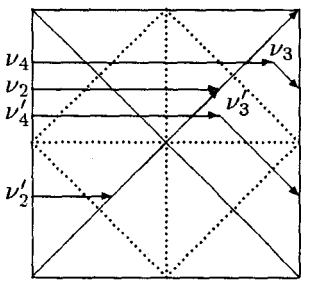
The formula φ_2 corresponds to the configurations that can reach some configuration at ℓ_4 satisfying ϕ_1 (hence, satisfying $y - x = 0$), and all the intermediary configurations satisfy ϕ_0 (hence, they must satisfy $y + x \leq 1$).

Consider the valuations $\nu_2 = (0, 0.7)$ and $\nu_2' = (0, 0.3)$. It is clear that $\nu_2 \simeq_2 \nu_2'$. However, it can be seen that $\langle \ell_1, \nu_2\rangle$ does not satisfy φ_2 since in order to reach a configuration at ℓ_4 which satisfies ϕ_1, we have to wait at ℓ_1 until the point $(0.7, 0.7)$ which does not satisfy ϕ_0. On the other hand, it is easy to see that $\langle \ell_1, \nu_2'\rangle$ satisfies φ_2.

Now, let us consider the formula φ_3. This formula is satisfied by the configurations at location ℓ_0 from which it is possible to reach ℓ_1 with some configuration

that satisfies φ_2. Notice, that in the definition of φ_3 above, we use (the path formula) ψ_2 instead of (the state formula) φ_2; it can be seen that this gives an equivalent formula which is in CTL_2^* whereas the use of φ_2 gives (unnecessarily) a CTL_3^* formula. Then, consider the valuations $\nu_3 = (0.9, 0.8)$ and $\nu_3' = (0.7, 0.6)$. These valuations are $\simeq_3$-equivalent, however, it can be seen that $\langle \ell_0, \nu_3 \rangle$ does not satisfy φ_3 whereas $\langle \ell_0, \nu_3' \rangle$ does.

Finally, let us consider the formula φ_4. It is satisfied by configurations which are at ℓ_1, and by staying at ℓ_1 until reaching some point where ϕ_1 is false, i.e., where $x > y$, they can reach some configuration which satisfies φ_3. Then, consider the valuations $\nu_4 = (0, 0.8)$ and $\nu_4' = (0, 0.6)$. Again, it can be seen that these valuations are $\simeq_4$-equivalent but the configurations $\langle \ell_1, \nu_4 \rangle$ and $\langle \ell_1, \nu_4' \rangle$ are distinguished by φ_4. It is easy to see that this process can be repeated forever. The following picture illustrates the discussion above.



Proposition 8.2 *$\forall n \geq 2$, there exist a location ℓ in $\mathcal{S}$ (ℓ_0 or ℓ_1) and two valuations ν_n and ν_n' such that $\nu_n \simeq_n \nu_n'$, $\langle \ell, \nu_n \rangle \notin [\![\varphi_n]\!]$, and $\langle \ell, \nu_n' \rangle \in [\![\varphi_n]\!]$.*

It can be seen that for every two points in $[0,1]^2$, there exists n such that these points are distinguished by $\simeq_n$. Hence, any finite partition of $[0,1]^2$ is included in some $\simeq_n$. Thus, a consequence of Proposition 8.2 and Proposition 5.1 is that there is no boundedly finite equivalence on valuations which can induce a bisimulation on the configurations of $\mathcal{S}$.

Theorem 8.1 *There exists a single-clock 2-dim EIG $\mathcal{S}$ such that, for every boundedly finite equivalence $\cong$ on valuations, there exists a location ℓ in $\mathcal{S}$ and two valuations ν and ν' such that $\nu \cong \nu'$ and $\langle \ell, \nu \rangle \not\sim \langle \ell, \nu' \rangle$.*

9 The Verification Problem

In this section, we present decidability results for the verification problem of systems we have considered in the previous sections. We prove mainly that the verification problem of ω-regular properties (ECTL_1^* formulas) for single-integrator 2-dim EIG's is decidable. For that, we show that this problem can be reduced to the inclusion problem of 1-counter ω-context-free languages in ω-regular ones.

Let us start with the simple case of 2-dim IG's. We show that in this case, the verification problem of all ECTL* formulas is decidable. Indeed, by Proposition 7.1, verifying ECTL* formulas for such a system $\mathcal{H}$ can be done by a

considering the *discrete* transition graph $\mathcal{R}_{\mathcal{H}}$ obtained as the quotient of the *dense* transition graph $\mathcal{G}_{\mathcal{H}}$ w.r.t. the equivalence on configurations induced by $\approx$. Moreover, since the equivalence $\approx$ is finite-index the graph $\mathcal{R}_{\mathcal{H}}$ is finite, it corresponds exactly to the *region graph* which is used in [1] for the verification of timed graphs. However, all the paths in $\mathcal{R}_{\mathcal{H}}$ does not correspond necessarily to computation sequence since the latters must be generated by time-diverging timed computations. Therefore, we define a path constraint (ω-regular property) *nonZeno* which is satisified by a path of $\mathcal{R}_{\mathcal{H}}$ if and only if it corresponds to a computation sequence. The definition of *nonZeno* is explained later in the more general case of single-integrator 2-dim EIG's. Then, given a formula φ of ECTL*, we consider the formula φ_{nz} which is obtained from φ by replacing recursively each subformula of the form $\exists\Omega$ by the formula $\exists(nonZeno \cap \Omega)$. Then, a configuration $\langle \ell, \nu \rangle$ of $\mathcal{H}$ satisfies the formula φ if and only if the node $\langle \ell, [\nu]_{\approx} \rangle$ of $\mathcal{R}_{\mathcal{H}}$, where $[\nu]_{\approx}$ is the $\approx$-equivalence class of ν, satisfies the formulas φ_{nz}. Since the verification problem of ECTL* for finite-state systems is decidable, we deduce the following fact.

Proposition 9.1 *The verification problem of 2-dim IG's w.r.t. ECTL* formulas is decidable.*

Now, let us consider the case of single-integrator 2-dim EIG's, and let $\mathcal{H}$ be such a system. As in the previous case, by Proposition 8.1, reasoning about ω-regular properties (or ECTL_1^* properties) of single-integrator 2-dim EIG's can be done by considering the quotient of the transition graph w.r.t. the equivalence induced by $\simeq_2$. However, by contrast with the case of 2-dim IG's, this quotient is *infinite* because $\simeq_2$ has an infinite number of classes. Nevetheless, this graph can fold up on a 1-counter automaton $\mathcal{A}_{\mathcal{H}}$, the counter represents the integer part of the $\{-1, 0, 1\}$-variable y (i.e., $\lfloor y \rfloor$), and the finite set of control nodes corresponds to the following informations: a control location of $\mathcal{H}$, is $fract(y) = 0$ or not, is $x > c_x$ or not, and in case $x \leq c_x$, the interval $[n, n]$ or $(n, n+1)$ (with integer bounds) which contains x, and how $fract(x) - fract(y)$ and $fract(x) + fract(y)$ are compared with 0 and 1 respectively.

It remains to guaranty that only paths corresponding to time diverging computations are considered. For that, we introduce a Muller acceptance condition which is described below.

First of all, we consider additional atomic propositions in order to take into account, in the definition of the control nodes, of the additional information whether a node corresponds to a $\simeq_1$-boundary region and is the target of only time progress transitions from nodes with non $\simeq_1$-boundary regions. Nodes satisfying this condition allow to observe the progress of time. Thus, a path which visits infinitely often such a node must be accepted, and hence, any set which contains such a node is considered as a repetition set.

However, there are two other kinds of paths that correspond to time diverging computations. Indeed, there are time-diverging computations that do not cross infinitly many times $\simeq_1$-boundary regions using time progress transitions.

The first kind of path we must accept are those corresponding to computations that, after some point, stay forever at some unbounded $\simeq_2$ equivalence

class, i.e., where $x > c_x$ and y remains in some open interval $(n, n+1)$. For that, we consider as repetition set any strongly connected set of nodes in the transition graph of the automaton $\mathcal{A}_\mathcal{H}$ such that all its nodes correspond to a same such an unbounded class, and where, either there is some node correponding to a location ℓ with $\partial(\ell, y) = 0$, or there are two nodes corresponding to locations ℓ_1 and ℓ_2 with $\partial(\ell_1, y) = 1$ and $\partial(\ell_2, y) = -1$. In both cases, we can indeed construct a time diverging computation where from some point, the value of y is either always the same (using the node where its rate is 0), or oscillates between two fixed values that are small enough to stay in the same class (this is possible since the considered class is open).

The other kind of paths we must accept are those corresponding for instance to computations that, after some point, go from some configuration $\langle \ell, \nu \rangle$ such that $\nu(x) = 0$ to a configuration $\langle \ell', \nu' \rangle$ where $\nu(y) = \nu'(y)$ without visiting any $\simeq_1$-boundary region, and then, reset the variable x and return back to the configuration $\langle \ell, \nu \rangle$. Then, we consider as repetition set any strongly connected set of nodes N in the transition graph of $\mathcal{A}_\mathcal{H}$ such that there exists a transition between two nodes in N that reset x, and either there exists some node in N corresponding to a location ℓ with $\partial(\ell, y) = 0$, or there are two nodes in N corresponding to locations ℓ_1 and ℓ_2 with $\partial(\ell_1, y) = 1$ and $\partial(\ell_2, y) = -1$. We consider also the case where y is reset, and there is a node in N with $\partial(\ell, x) = 0$. It can be seen that in such cases, we can construct a time diverging computation which takes at each cycle, some fixed amount of time ϵ.

Notice that the Muller condition defined above can be encoded, using additional atomic propositions (one for each repetition set) as an ω-regular property which corresponds to the property *nonZeno* mentioned above in the case of 2-dim IG's.

Then, a configuration $\langle \ell, \nu \rangle$ satisfies an ω-regular property Ω, i.e., $\langle \ell, \nu \rangle \in [\![\forall \Omega]\!]$ iff the ω-context-free language recognized by $\mathcal{A}_\mathcal{H}$ starting from $\langle \ell, [\nu]_{\simeq_2} \rangle$ is included in Ω. Since the inclusion problem between ω-context-free languages and ω-regular ones is decidable, we deduce that the verification problem of ω-regular properties for single-integrator 2-dim EIG's is decidable. Then, by the fact that ECTL_1^* formulas are boolean combinations of formulas of the form $\forall\Omega$, we obtain the following decidability result.

Theorem 9.1 *The verification problem of single-integrator 2-dim EIG's w.r.t. $ECTL_1^*$ formulas is decidable.*

10 Conclusion

We have presented a decidability result for a subclass of linear hybrid systems that are 2-dim EIG with one integrator and one $\{-1, 0, 1\}$-variable. We have established this result by showing that these systems generate ω-context-free sets of state sequences.

Our work shows that there are hybrid systems whose verification problem is decidable, but cannot be reduced to a verification problem on finite-state

systems, as it is usually the case. It shows also the relevance of the works on infinite-state systems to the verification of hybrid systems.

Moreover, this work shows that there are hybrid systems whose linear-time properties can be verified whereas there is no (boundedly) finite bisimulation on their set of configurations. The systems we consider are always trace equivalent to pushdown automata, but they are not necessarily bisimilar to them. Hence, these systems cannot be handled using a finite bisimulations based approach [9].

Acknowledgment We thank T. Henzinger, P. Kopke, and Y. Lakhnech for interesting discussions and judicious comments.

References

1. R. Alur, C. Courcoubetis, and D. Dill. Model-Checking for Real-Time Systems. In *LICS'90*. IEEE, 1990.
2. 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*. LNCS 736, 1993.
3. E. Asarin, O. Maler, and A. Pnueli. Reachability Analysis of Dynamical Systems Having Piecewise-Constant Derivatives. *T.C.S.*, 138, 1995.
4. Z. Chaochen, C.A.R. Hoare, and A.P. Ravn. A Calculus of Durations. *Information Processing Letters*, 40:269–276, 1991.
5. E. Clarke, A. Emerson, and P. Sistla. Automatic Verification of Finite State Concurrent Systems using Temporal Logic Specifications: A Practical Approach. In *POPL'83*, 1983.
6. R.S. Cohen and A.Y. Gold. Theory of ω-Languages. I: Characterizations of ω-Context-Free Languages. *J.C.S.S.*, 15:169–184, 1977.
7. E.Clarke, O. Grümberg, and R. Kurshan. A Synthesis of two Approaches for Verifying Finite State Concurrent Systems. In *CMU tech. rep.*, 1987.
8. E.A. Emerson and J. Y. Halpern. 'Sometimes' and 'Not Never' Revisited: On Branching vs. Linear Time Logic. In *POPL'83*, 1983.
9. T. Henzinger. Hybrid Automata with Finite Bisimulations. In *ICALP'95*, 1995.
10. T.A. Henzinger, P.W. Kopke, A. Puri, and P. Varaiya. What's Decidable about Hybrid Automata. In *STOC'95*, 1995.
11. Y. Kesten, A. Pnueli, J. Sifakis, and S. Yovine. Integration Graphs: A Class of Decidable Hybrid Systems. In *Hybrid Systems*. LNCS 736, 1993.
12. O. Maler, Z. Manna, and A. Pnueli. From Timed to Hybrid Systems. In *REX workshop on Real-Time: Theory and Practice*. LNCS 600, 1992.
13. D.E. Muller. Infinite Sequences and Finite Machines. In *4th Symp. on Switching Circuit Theory and Logical Design*. IEEE, 1963.
14. X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine. An Approach to the Description and Analysis of Hybrid Systems. In *Hybrid Systems*. LNCS 736, 1993.
15. A. Pnueli. The Temporal Logic of Programs. In *FOCS'77*. IEEE, 1977.
16. W. Thomas. Computation Tree Logic and Regular ω-Languages. LNCS 354, 1989.
17. W. Thomas. Automata on Infinite Objects. In *Handbook of Theo. Comp. Sci.* Elsevier Sci. Pub., 1990.
18. M.Y. Vardi. A Temporal Fixpoint Calculus. In *POPL'88*, 1988.
19. P. Wolper. Temporal Logic Can Be More Expressive. *Inform. and Cont.*, 56, 1983.