Logical Methods in Computer Science
Vol. 6 (1:1) 2010, pp. 1–48
www.lmcs-online.org

Submitted Apr. 9, 2007
Published Jan. 12, 2010

O-MINIMAL HYBRID REACHABILITY GAMES

PATRICIA BOUYER [a], THOMAS BRIHAYE [b], AND FABRICE CHEVALIER [a]

[a] LSV, CNRS & ENS Cachan, 61, avenue du Président Wilson, 94230 Cachan, France
e-mail address: {bouyer,chevalie}@lsv.ens-cachan.fr

[a] Université de Mons, 20, place du parc, 7000 Mons, Belgium
e-mail address: thomas.brihaye@umons.ac.be

Abstract. In this paper, we consider reachability games over general hybrid systems, and distinguish between two possible observation frameworks for those games: either the precise dynamics of the system is seen by the players (this is the perfect observation framework), or only the starting point and the delays are known by the players (this is the partial observation framework). In the first more classical framework, we show that time-abstract bisimulation is not adequate for solving this problem, although it is sufficient in the case of timed automata. That is why we consider an other equivalence, namely the suffix equivalence based on the encoding of trajectories through words. We show that this suffix equivalence is in general a correct abstraction for games. We apply this result to o-minimal hybrid systems, and get decidability and computability results in this framework. For the second framework which assumes a partial observation of the dynamics of the system, we propose another abstraction, called the superword encoding, which is suitable to solve the games under that assumption. In that framework, we also provide decidability and computability results.

1. Introduction

Games over hybrid systems. Hybrid systems are finite-state machines equipped with a continuous dynamics. In the last thirty years, formal verification of such systems has become a very active field of research in computer science, with numerous success stories. In this context, hybrid automata, an extension of timed automata [AD90, AD94], have been intensively studied [Hen95, Hen96], and decidable subclasses of hybrid systems have been drawn like initialized rectangular hybrid automata [Hen96]. More recently, games over hybrid systems have appeared as a new interesting and active field of research since, among others, they correspond to a formulation of control problems, the counterpart of model checking for open systems, *i.e.*, systems embedded in a possibly reactive environment. In this context, many results have already been obtained, like the (un)decidability of control problems for hybrid automata [HHM99], or (semi-)algorithms for solving such problems [dAHM01].

1998 ACM Subject Classification: F.3.1, F.4.1.
Key words and phrases: O-minimal hybrid systems, Reachability games, Synthesis.

DOI:10.2168/LMCS-6 (1:1) 2010

Given a system S (with controllable and uncontrollable actions) and a property φ, controlling the system means building another system C (which can only enforce controllable actions), called the controller, such that $S \parallel C$ (the system S guided by the controller C) satisfies the property φ. In our context, the property is a reachability property and our aim is to build a controller enforcing a given location of the system, whatever the environment does (which plays with the uncontrollable actions).

O-minimal hybrid systems. O-minimal hybrid systems have been first proposed in [LPS00] as an interesting class of systems (see [vdD98] for an overview of properties of o-minimal structures). They have very rich continuous dynamics, but limited discrete steps (at each discrete step, all variables have to be reset, independently from their initial values). This allows to decouple the continuous and discrete components of the hybrid system (see [LPS00]). Thus, properties of a global o-minimal system can be deduced directly from properties of the continuous parts of the system. Since the introductory paper [LPS00], several works have considered o-minimal hybrid systems [Dav99, BMRT04, BM05, KV04, KV06], mostly focusing on abstractions of such systems, on reachability properties, and on bisimulation properties.

Word encoding. In [BMRT04], an encoding of trajectories with words has been proposed in order to prove the existence of finite bisimulations for o-minimal hybrid systems (see also [BM05]). Let us mention that this technique has been used in [KV04, KV06] in order to provide an exponential bound on the size of the finite bisimulation in the case of pfaffian hybrid systems. Let us also notice that similar techniques already appeared in the literature, see for instance the notion of signature in [ASY01]. Different word encoding techniques have been studied in a wider context in [Bri07]. Recently in [KRS07], the authors propose a new algorithm for counter-example guided abstraction and refinement on hybrid systems, based on use a word encoding approach. In this paper we use the so-called *suffix encoding*, which was shown to be in general too fine to provide the coarsest time-abstract bisimulation. However, based on this encoding, a semi-algorithm has been proposed in [Bri07, Bri06] for computing a time-abstract bisimulation, and it terminates in the case of o-minimal hybrid systems.

Contributions of this paper. In this paper, we focus on games over hybrid systems. We describe two rather natural frameworks for such games, one assuming a perfect observation of the dynamics of the system, and another one assuming a partial observation of the dynamics. For the first framework, we use the above-mentioned suffix word encoding of trajectories for giving sufficient computability conditions for the winning states of a game. Time-abstract bisimulation is an equivalence relation which is correct with respect to reachability properties on hybrid systems [AHLP00] and with respect to control reachability properties on timed automata [AMPS98]. Here, we show that the time-abstract bisimulation is not correct anymore for solving control problems on a general class of hybrid systems: we exhibit a system in which two states are time-abstract bisimilar, but one of the states is winning and the other is not. Using the suffix encoding of trajectories of [Bri07], we prove that, in the perfect observation framework, two states having the same suffixes are equivalently winning or losing (this is a stronger condition than the one for the time-abstract bisimulation). We then focus on o-minimal hybrid games and prove that, under the assumption that the theory of the underlying o-minimal structure is decidable, the control problem can be solved and that winning states and winning strategies can be computed. Regarding the partial observation framework, we provide a new encoding technique, the so-called super-word encoding, which turns out to be sound for the control under partial observation of the dynamics, and which allows to prove decidability and computability results similar to those in the perfect observation framework.

Related work. The most relevant related works are those dealing with hybrid games [HHM99, dAHM01]. However, the framework of these papers is pretty different from ours:

(1) In their framework, time is considered as a discrete action, and once action "let time elapse" has been chosen, it is not possible to bound the time elapsing, which is quite restrictive. For instance, the timed game of Figure 1 is winning from $(\ell_0, x = 0)$ in our framework (the strategy is to wait some amount of time $t \in [2, 5]$ and to take the controllable action c), whereas it is not winning in their framework (once x is above 5, it is no more possible to take the transition and reach the winning location ℓ_1, and there is no way to impose a delay within $[2, 5]$). This yields significant differences in the properties: in their framework, game bisimulation is one of the tools for solving the games, and as stated by [HHM99, Prop. 1], the classical bisimulation tool is then sufficient to solve games. On the contrary, in our framework, the notion of bisimulation relevant to our model (time-abstract bisimulation) is not correct for solving games, as will be explored in this paper.

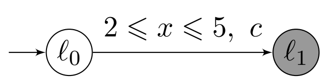


Figure 1: A simple game

(2) Our games are control games, they are thus asymmetric, which is not the case of the games in the above-mentioned works; in our framework, the environment is more powerful than the controller in that it can outstrip the controller and do an action right before the controller decides to do a controllable action.

Let us also mention the paper [WT97] on control of linear hybrid automata. In [WT97] the author proposes a semidecision procedure for synthesizing controllers for such automata. No general decidability result is given in this paper.

Plan of the paper. In Section 2, we recall results about finite games and bisimulation. In Section 3, we define the games over dynamical systems (for both perfect information and partial observation), and we show that time-abstract bisimulation is not correct for solving them. The word encoding techniques are presented in Section 4 and used in Section 5 to present a general framework for solving games over dynamical systems. We apply and extend these results in Section 6 for computing winning states and winning strategies in o-minimal games. In the paper, we often only develop technical details of the partial observation framework, which actually extends the perfect observation framework.

Part of the results presented in this paper have been published in [BBC06] (the decidability of the control reachability problem and the synthesis of strategies for o-minimal hybrid systems). In this paper, we give full proofs of those results, and extend them to a natural partial observation framework.

2. Classical Finite Games

In this section, we recall some basic definitions and results concerning bisimulations on a transition system (see [Acz88, Mil89, Cau95, Hen95] for general references) and classical (untimed) games.

2.1. **Classical Games.**

We present here the definitions of the problem of control on a finite graph (also called finite game) and the notion of strategy (see [GTW02] for an overview on games). These definitions are classical and will be extended to real-time systems in the next section.

Definition 2.1. A *finite automaton* is a tuple $\mathcal{A} = (Q, \mathsf{Goal}, \Sigma, \delta)$ where Q is a finite set of locations, $\mathsf{Goal} \subseteq Q$ is a subset of winning locations, Σ is a finite set of actions, and δ consists of a finite number of transitions $(q, a, q') \in Q \times \Sigma \times Q$.

Definition 2.2. A *transition system* $T = (Q, \Sigma, \to)$ consists of a set of states Q (which may be uncountable), Σ an alphabet of events, and $\to \subseteq Q \times \Sigma \times Q$ a transition relation.

A transition $(q_1, a, q_2) \in \to$ is also denoted by $q_1 \xrightarrow{a} q_2$. A transition system is said finite if Q is finite. Note that a finite automaton canonically defines a transition system $T_{\mathcal{A}}$.

A *run* of $\mathcal{A}$ is a finite or infinite sequence $q_0 \xrightarrow{a_1} q_1 \xrightarrow{a_2} \dots$ of the transition system $T_{\mathcal{A}}$. Such a run is said *winning* if $q_i \in \mathsf{Goal}$ for some i. If ρ is a finite run $q_0 \xrightarrow{a_1} q_1 \xrightarrow{a_2} \dots \xrightarrow{a_n} q_n$ we define $last(\rho) = q_n$. We note $\mathsf{Runs}_f(\mathcal{A})$ the set of finite runs in $\mathcal{A}$.

Definition 2.3. A *finite game* is a finite automaton $(Q, \mathsf{Goal}, \Sigma, \delta)$ where Σ is partitioned into two subsets Σ_c and Σ_u corresponding to controllable and uncontrollable actions.

We will consider *control games*. Informally there are two players in such a game: the *controller* and the *environment*. The actions of Σ_c belong to the controller and the actions of Σ_u belong to the environment. At each step, the controller proposes a controllable action which corresponds to the action he wants to perform; then either this action or an uncontrollable action is done and the automaton goes into one of the next states[1]. In the sequel, we will only consider reachability games : the controller wants to reach the Goal states and the environment wants to prevent him from doing so.

Definition 2.4. A *strategy* is a partial function λ from $\mathsf{Runs}_f(\mathcal{A})$ to Σ_c such that for all runs $\rho \in \mathsf{Runs}_f(\mathcal{A})$, if $\lambda(\rho)$ is defined, then it is enabled in $last(\rho)$.

Let $\rho = q_0 \xrightarrow{a_1} q_1 \xrightarrow{a_2} \dots$ be a run, and set for every i, ρ_i the prefix of length i of ρ. The run ρ is said *compatible with a strategy* λ when for all i, $a_{i+1} = \lambda(\rho_i)$ or $a_{i+1} \in \Sigma_u$. A run ρ is said *maximal w.r.t. a strategy* λ if it is infinite or if $\lambda(\rho)$ is not defined.

A strategy λ is *winning from a state* q if all maximal runs starting in q compatible with λ are winning.

[1] There may be several next states as the game is not supposed to be deterministic, and we assume that the environment chooses the next state in case there are several.

2.2. **Bisimulation.** We recall now the definition of bisimulation for transition systems:

Definition 2.5 ([Mil89, Cau95])**.** Given a transition system $T = (Q, \Sigma, \rightarrow)$, a *bisimulation for* T is an equivalence relation $\sim \subseteq Q \times Q$ such that $\forall q_1, q_1', q_2 \in Q$, $\forall a \in \Sigma$,

$$\Big(q_1 \sim q_1' \text{ and } q_1 \xrightarrow{a} q_2\Big) \Rightarrow \Big(\exists q_2' \; q_2 \sim q_2' \;\text{ and }\; q_1' \xrightarrow{a} q_2'\Big)$$

Moreover, if $\mathcal{P}$ is a partition of Q and if $\sim$ respects $\mathcal{P}$ (*i.e.*, $q \in P$ and $q \sim q'$ with $P \in \mathcal{P}$ implies $q' \in P$), we say that $\sim$ is *compatible* with $\mathcal{P}$.

2.3. **Game and Bisimulation in the Untimed Case.** In the untimed framework, bisimulation is a commonly used technique to abstract games: bisimilar states can be identified in the control problem. This is stated in the next folklore theorem, for which we provide a proof.

Theorem 2.6. *Let $\mathcal{A} = (Q, \mathsf{Goal}, \Sigma, \delta)$ be a finite game, $q, q' \in Q$ and $\sim$ a bisimulation compatible with* Goal*. Then, there is a winning strategy from q iff there is a winning strategy from q'.*

Proof. Assume that $\sim$ is a bisimulation relation compatible with Goal and such that $q \sim q'$. Assume furthermore that λ is a winning strategy from q. We will define a strategy λ' that will be winning from q'. To do that we will map finite runs starting in q' to finite runs starting in q, so that λ' will mimick λ through this mapping. We note f this mapping, and start by setting $f(q') = q$. We then proceed inductively as follows. If $\lambda(f(\varrho'))$ is defined, we set $\lambda'(\varrho') = \lambda(f(\varrho'))$ and for every run $\varrho' \xrightarrow{\lambda'(\varrho'))} \widetilde{q}'$ (which is compatible with λ') there is a run $f(\varrho') \xrightarrow{\lambda(\varrho)} \widetilde{q}$ which is compatible with λ and such that $\widetilde{q} \sim \widetilde{q}'$. We then define $f(\varrho' \xrightarrow{\lambda'(\varrho')} \widetilde{q}') = f(\varrho') \xrightarrow{\lambda(\varrho)} \widetilde{q}$. The strategy λ' is winning from q' since $\sim$ is compatible with Goal. □

This theorem remains true for infinite-state discrete games [HHM99, dAHM01] and can be used to solve them: if an infinite-state game has a bisimulation of finite index, the control problem can be reduced to a control problem over a finite graph. Real-time control problems cannot be seen as classical infinite-state games because of the special nature of the time-elapsing action. which does not belong to one of the players. It seems nevertheless natural to try to adapt the bisimulation approach to solve real-time control problems.

3. Games over Dynamical Systems

3.1. **Dynamical Systems.** Let $\mathcal{M}$ be a structure. When we say that some relation, subset or function is *definable*, we mean it is first-order definable in the structure $\mathcal{M}$. A general reference for first-order logic is [Hod97]. We denote by $\mathsf{Th}(\mathcal{M})$ the theory of $\mathcal{M}$. In this paper we only consider structures $\mathcal{M}$ that are expansions of ordered groups, we also assume that the structure $\mathcal{M}$ contains two symbols of constants, *i.e.*, $\mathcal{M} = \langle M, +, 0, 1, <, \ldots \rangle$ where $+$ is the group operation and w.l.o.g. we assume that $0 < 1$.

Definition 3.1. A *dynamical system* is a pair $(\mathcal{M}, \gamma)$ where:

- $\mathcal{M} = \langle M, +, 0, 1, <, \ldots \rangle$ is an expansion of an ordered group,

- $\gamma : V_1 \times V \to V_2$ is a function definable in $\mathcal{M}$ (where $V_1 \subseteq M^{k_1}$, $V \subseteq M$ and $V_2 \subseteq M^{k_2}$).[2]

The function γ is called the *dynamics* of the system.

Classically, when M is the field of the reals, we see V as the time, V_1 as the input space, $V_1 \times V$ as the space-time and V_2 as the (output) space. We keep this terminology in the more general context of a structure $\mathcal{M}$.

The definition of *dynamical system* encompasses a lot of different behaviors. Let us first give a simple example, several others will be presented later.

Example 3.2. We can recover the continuous dynamics of *timed automata* (see [AD94]). In this case, we have that $\mathcal{M} = \langle \mathbb{R}, <, +, 0, 1\rangle$ and the dynamics $\gamma : \mathbb{R}^n \times [0, +\infty[\to \mathbb{R}^n$ is defined by $\gamma(x_1, \dots, x_n, t) = (x_1 + t, \dots, x_n + t)$.

Definition 3.3. If we fix a point $x \in V_1$, the set $\Gamma_x = \{\gamma(x, t) \mid t \in M^+\} \subseteq V_2$ is called the *trajectory* determined by x.

We define a transition system associated with the dynamical system. This definition is an adaptation to our context of the classical *continuous transition system* in the case of hybrid systems (see [LPS00] for example).

Definition 3.4. Given $(\mathcal{M}, \gamma)$ a dynamical system, we define a *transition system* $T_\gamma = (Q, \Sigma, \to_\gamma)$ *associated with the dynamical system* by:

- the set Q of states is V_2;
- the set Σ of events is $M^+ = \{\tau \in M \mid \tau \geqslant 0\}$;
- the transition relation $y_1 \xrightarrow{t}_\gamma y_2$ is defined by:

$$\begin{aligned} &\exists x \in V_1,\ \exists t_1, t_2 \in M^+ \text{ such that } t_1 \leqslant t_2, \\ &\quad \gamma(x, t_1) = y_1,\ \gamma(x, t_2) = y_2 \text{ and } t = t_2 - t_1 \end{aligned}$$

3.2. **$\mathcal{M}$-Games Under Perfect Observation.** In this subsection, we define $\mathcal{M}$-automata, which are automata with guards, resets and continuous dynamics definable in the $\mathcal{M}$-structure. We then introduce our model of dynamical game which is an $\mathcal{M}$-automaton with two sets of actions, one for each player; we finally express in terms of winning strategy the main problem we will be interested in, the control problem in a class $\mathcal{C}$ of $\mathcal{M}$-automata under perfect observation. The partial observation framework will be discussed in Subsection 3.3.

Definition 3.5 ($\mathcal{M}$-automaton)**.** An $\mathcal{M}$-*automaton* $\mathcal{A}$ is a tuple $(\mathcal{M}, Q, \mathsf{Goal}, \Sigma, \delta, \gamma)$ where $\mathcal{M} = \langle M, +, 0, 1, <, \dots\rangle$ is an expansion of an ordered group, Q is a finite set of locations, $\mathsf{Goal} \subseteq Q$ is a subset of winning locations, Σ is a finite set of actions, δ consists in a finite number of transitions $(q, g, a, R, q') \in Q \times 2^{V_2} \times \Sigma \times (V_2 \to 2^{V_2}) \times Q$ where g and R are definable in $\mathcal{M}$, and γ maps every location $q \in Q$ to a dynamics $\gamma_q : V_1 \times V \to V_2$.

We use a general definition for resets: a reset R is indeed a general function from V_2 to 2^{V_2}, which may correspond to a non-deterministic update. If the current state is (q, y) the system will jump to some (q', y') with $y' \in R(y)$.

An $\mathcal{M}$-automaton $\mathcal{A} = (\mathcal{M}, Q, \mathsf{Goal}, \Sigma, \delta, \gamma)$ defines a *mixed transition system* $T_{\mathcal{A}} = (S, \Gamma, \to)$ where:

[2]We use these notations in the rest of the paper.

- the set S of states is $Q \times V_2$;
- the set Γ of labels is $M^+ \cup \Sigma$, (where $M^+ = \{\tau \in M \mid \tau \geqslant 0\}$);
- the transition relation $(q,y) \xrightarrow{e} (q',y')$ is defined when:
  - $e \in \Sigma$, and there exists $(q,g,e,R,q') \in \delta$ with $y \in g$ and $y' \in R(y)$, or
  - $e \in M^+$, $q = q'$, and $y \xrightarrow{e}_{\gamma_q} y'$ where γ_q is the dynamic in location q.

In the sequel, we will focus on behaviors of $\mathcal{M}$-automata which alternate between continuous transitions and discrete transitions.

We will also need more precise notions of transitions. When $(q,y) \xrightarrow{\tau} (q,y')$ with $\tau \in M^+$, this is due to some choice of $(x,t) \in V_1 \times V$ such that $\gamma_q(x,t) = y$. We say that $(q,y) \xrightarrow{\tau}_{x,t} (q,y')$ if $\gamma_q(x,t) = y$ and $\gamma_q(x,t+\tau) = y'$. To ease the reading of the paper, we will sometimes write $(q,x,t,y) \xrightarrow{\tau} (q,x,t+\tau,y')$ for $(q,y) \xrightarrow{\tau}_{x,t} (q,y')$. We say that an action $(\tau,a) \in M^+ \times \Sigma$ is enabled in a state (q,x,t,y) if there exists (q',x',t',y') and (q'',x'',t'',y'') such that $(q,x,t,y) \xrightarrow{\tau} (q',x',t',y') \xrightarrow{a} (q'',x'',t'',y'')$. We then write $(q,x,t,y) \xrightarrow{\tau,a} (q'',x'',t'',y'')$.

A *run* of $\mathcal{A}$ is a finite or infinite sequence $(q_0,x_0,t_0,y_0) \xrightarrow{\tau_1,a_1} (q_1,x_1,t_1,y_1)\ldots$ Such a run is said *winning* if $q_i \in \mathsf{Goal}$ for some i.

We note $\mathsf{Runs}_f(\mathcal{A})$ the set of finite runs in $\mathcal{A}$. If ρ is a finite run $(q_0,x_0,t_0,y_0) \xrightarrow{\tau_1,a_1} \ldots \xrightarrow{\tau_n,a_n} (q_n,x_n,t_n,y_n)$ we define $last(\rho) = (q_n,x_n,t_n,y_n)$.

Definition 3.6 ($\mathcal{M}$-game)**.** An $\mathcal{M}$-*game* is an $\mathcal{M}$-automaton $(\mathcal{M},Q,\mathsf{Goal},\Sigma,\delta,\gamma)$ where Σ is partitioned into two subsets Σ_c and Σ_u corresponding to controllable and uncontrollable actions.

Definition 3.7 (Strategy)**.** A *strategy*[3] is a partial function λ from $\mathsf{Runs}_f(\mathcal{A})$ to $M^+ \times \Sigma_c$ such that for all runs ρ in $\mathsf{Runs}_f(\mathcal{A})$, if $\lambda(\rho)$ is defined, then it is enabled in $last(\rho)$.

The strategy tells what is to be done at the current moment: at each instant it tells what delay we will wait and which controllable action will be taken after this delay. Note that the environment may have to choose between several edges, each labeled by the action given by the strategy (because the original game is not supposed to be deterministic).

A strategy λ is said *memoryless* if for all finite runs ρ and ρ', $last(\rho) = last(\rho')$ implies $\lambda(\rho) = \lambda(\rho')$. Let $\rho = (q_0,x_0,t_0,y_0) \xrightarrow{\tau_1,a_1} \ldots$ be a run, and set for every i, ρ_i the prefix of length i of ρ. The run ρ is said *consistent with a strategy* λ when for all i, if $\lambda(\rho_i) = (\tau,a)$ then either $\tau_{i+1} = \tau$ and $a_{i+1} = a$, or $\tau_{i+1} \leqslant \tau$ and $a_{i+1} \in \Sigma_u$. A run ρ is said *maximal w.r.t. a strategy* λ if it is infinite or if $\lambda(\rho)$ is not defined. A strategy λ is *winning from a state (q,y)* if for all (x,t) such that $\gamma(x,t) = y$, all maximal runs starting in (q,x,t,y) compatible with λ are winning. The *set of winning states* is the set of states from which there is a winning strategy.

We can now define the control problems we will study.

Problem 3.8 (Control problem under perfect observation in a class $\mathcal{C}$ of $\mathcal{M}$-automata)**.** Given an $\mathcal{M}$-game $\mathcal{A} \in \mathcal{C}$, and a definable initial state (q,y), determine whether there exists a winning strategy in $\mathcal{A}$ from (q,y).

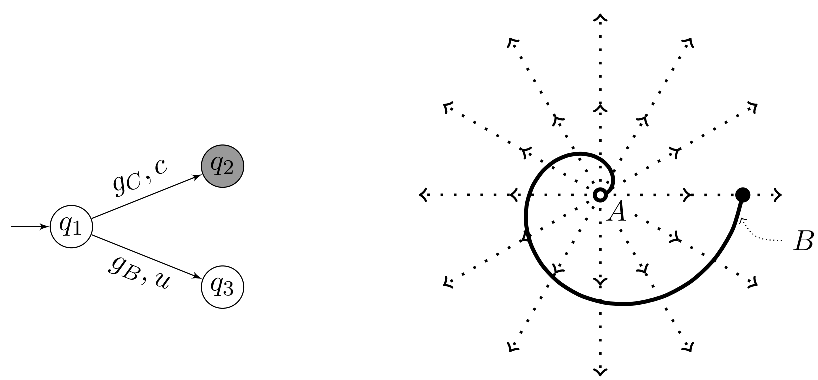
[3]In the context of control problems, a strategy is also called a *controller*.

Problem 3.9 (Controller synthesis under perfect observation in a class $\mathcal{C}$ of $\mathcal{M}$-automata)**.** Given an $\mathcal{M}$-game $\mathcal{A} \in \mathcal{C}$, and a definable initial state (q, y), determine whether there exists a winning strategy, and compute such a strategy if possible.[4]

Example 3.10. Let us consider the $\mathcal{M}$-game $\mathcal{A} = (\mathcal{M}, Q, \mathsf{Goal}, \Sigma, \delta, \gamma)$ (depicted in Fig. 2) where $\mathcal{M} = \langle \mathbb{R}, +, \cdot, 0, 1, <, \sin, \cos \rangle$, $Q = \{q_1, q_2, q_3\}$, $\mathsf{Goal} = \{q_2\}$, $\Sigma = \Sigma_c \cup \Sigma_u$ where $\Sigma_c = \{c\}$ (resp. $\Sigma_u = \{u\}$) is the set of controllable (resp. uncontrollable) actions. The dynamics in q_1, $\gamma_{q_1} : \mathbb{R}^2 \times [0, 2\pi] \times \mathbb{R} \to \mathbb{R}^2$ is defined as follows.

$$\gamma_{q_1}(x_1, x_2, \theta, t) = \begin{cases} (t.\cos(\theta), t.\sin(\theta)) & \text{if } (x_1, x_2) = (0, 0), \\ (x_1 + t.x_1, x_2 + t.x_2) & \text{if } (x_1, x_2) \neq (0, 0). \end{cases}$$

We associate with this dynamical system the partition $\mathcal{P} = \{A, B, C\}$ where $A = \{(0, 0)\}$, $B = \{(\theta\cos(\theta), \theta\sin(\theta)) \mid 0 < \theta \leqslant 2\pi\}$ and $C = \mathbb{R}^2 \setminus (A \cup C)$. Let us call piece B *the spiral* (see Figure 2(b)). The guard g_B corresponds to B-states (*i.e.*, points on the spiral) and the guard g_C corresponds to C-states (points not on the spiral and different from the origin). In this example, the point $(q_1, (0, 0))$ is a winning state. Indeed a winning strategy



(a) The $\mathcal{M}$-game $\mathcal{A}$ (b) Dynamics in q_1

Figure 2: Time-abstract bisimulation does not preserve winning states

is given by $\lambda(q_1, 0, 0, \theta, t) = (\frac{\theta}{2}, c)$ where c consists in taking the transition leading to state q_2 (which is winning).

3.3. **$\mathcal{M}$-Games Under Partial Observation.** Subsection 3.2, we have assumed that from a given point, the environment chooses the continuous trajectory followed by the game, and the controller reacts accordingly. In this section, we consider partial observation of the dynamics: the trajectory is not known by the controller, and its strategy may depend only on the current point. In particular, this framework naturally models drift of clocks where the slopes of the clocks lies within an interval [Pur98, ALM05]. Note that our partial observation assumption concerns the dynamics of the system, not the actions which are performed. This has to be contrasted with the notion of partial observation studied in the framework of finite systems in [AVW03] or in the context of timed systems in [BDMP03]

[4]In this definition, 'compute a strategy' means 'give a formula for the strategy'. In particular, a strategy which is computable is definable in the theory.

where the partial observation assumption concerns actions which are done, and not the dynamics (indeed, in these models, there is no real choice for the dynamics; It is completely determined by the point in the state-space). In order to formalize our partial observation framework, we need to adapt notions such as strategy in this new setting. First, we define what we call *observation* of a given run.

Definition 3.11 (Observation of a run)**.** Let $\rho = (q_0, x_0, t_0, y_0) \xrightarrow{\tau_1, a_1} \dots \xrightarrow{\tau_n, a_n} (q_n, x_n, t_n, y_n)$ be a finite run. The *observation* of ρ, denoted $\mathrm{obs}(\rho)$ is the sequence $(q_0, y_0) \xrightarrow{\tau_1, a_1} \dots \xrightarrow{\tau_n, a_n} (q_n, y_n)$.

Definition 3.12 (Strategy under partial observation)**.** A strategy λ is said *under partial observation* if for all finite runs ρ, ρ', $\mathrm{obs}(\rho) = \mathrm{obs}(\rho')$ implies $\lambda(\rho) = \lambda(\rho')$.

All other notions, like memoryless strategies, consistency, winning strategies, winning states, *etc...* naturally extend in this new context. In this setting, we will consider the two following problems.

Problem 3.13 (Control problem under partial observation in a class $\mathcal{C}$ of $\mathcal{M}$-automata)**.** Given an $\mathcal{M}$-game $\mathcal{A} \in \mathcal{C}$, and a definable initial state (q, y), determine whether there exists a winning strategy under partial observation in $\mathcal{A}$ from (q, y).

Problem 3.14 (Controller synthesis under partial observation in a class $\mathcal{C}$ of $\mathcal{M}$-automata)**.** Given an $\mathcal{M}$-game $\mathcal{A} \in \mathcal{C}$, and a definable initial state (q, y), determine whether there exists a winning strategy under partial observation in $\mathcal{A}$ from (q, y), and compute such a strategy if possible.

Example 3.15. We consider again the spiral example (Example 3.10). We showed that under perfect observation this $\mathcal{M}$-game has a winning strategy in $(q_1, (0, 0))$ given by $\lambda(q_1, 0, 0, \theta, t) = (\frac{\theta}{2}, c)$. Note that this strategy depends on the precise trajectory (parameter θ). Moreover, one can show that there is no winning strategy under partial observation for this game: such a strategy may only depend on the current point, and in this precise example, whatever action (τ, a) the controller proposes in $(q_1, (0, 0))$, there is a trajectory which reaches a *bad* state (*i.e.*, points on the spiral) before τ.

The previous example shows that some games can be winning under perfect observation whereas they are not winning under partial observation. Nevertheless, considering a new dynamics which will roughly inform the controller of the current trajectory, we can see the perfect observation control problem as a special case of the partial observation framework. This is stated by the following proposition :

Problem 3.16. Given an $\mathcal{M}$-game $\mathcal{A}_1$ and a state (q, y) of $\mathcal{A}_1$, we can effectively construct an $\mathcal{M}$-game $\mathcal{A}_2$ and a state (q', y') of $\mathcal{A}_2$ such that there exists a winning strategy under perfect observation in $\mathcal{A}_1$ from (q, y) iff there exists a winning strategy under partial observation in $\mathcal{A}_2$ from (q', y').

Proof. Let $\mathcal{A}_1 = (\mathcal{M}, Q, \mathsf{Goal}, \Sigma, \delta, \gamma)$ where $\gamma : V_1 \times V \to V_2$. We define $V_2' = \{(x, t, y) \in V_1 \times V \times V_2 \mid \gamma(x, t) = y\}$ and for $q \in Q$, $\gamma_q' : V_1 \times V \to V_2'$ such that $\gamma_q'(x, t) = (x, t, \gamma_q(x, t))$. The dynamics γ' behaves exactly like γ but "gives" to the controller the current trajectory as this information is stored in the state space V_2'.

We then use $\mathcal{A}_2 = (\mathcal{M}, Q, \mathsf{Goal}, \Sigma, \delta', \gamma')$, where δ' is the transition relation δ adapted to the new states V_2': if $(q_1, g, a, R, q_2) \in \delta$ then $(q_1, g', a, R', q_2) \in \delta'$ where $g' = \{(x, t, \gamma(x, t)) \mid \gamma(x, t) \in g\}$ and for all $(x, t) \in V_1 \times V$, $R'(\gamma(x, t)) = \{(x', t', \gamma(x', t')) \mid \gamma(x', t') \in R(\gamma(x, t))\}$.

W.l.o.g. we can suppose that there exists a unique $(x_0, t_0) \in V_1 \times V$ such that $\gamma(x_0, t_0) = y$ (if necessary, we add a location with constant continuous dynamics pointing to the actual location of y). Then there exists a winning strategy under perfect observation in $\mathcal{A}_1$ from (q, y) iff there exists a winning strategy under partial observation in $\mathcal{A}_2$ from $(q, (x_0, t_0, y))$. □

From the above proposition we get that any definability, decidability, *etc* result in the partial observation framework will hold in the perfect observation framework.

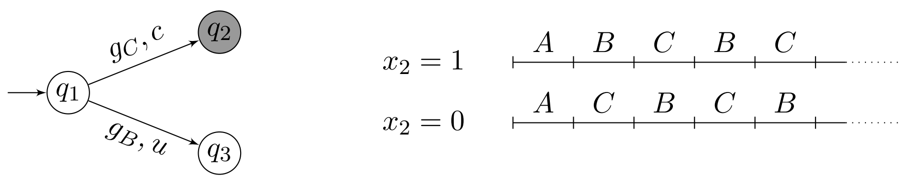
3.4. **$\mathcal{M}$-Games and Bisimulation.** Time-abstract bisimulation [Hen95, Dav99, AHLP00] is a sufficient behavioral relation to check reachability properties of hybrid systems, and in particular of $\mathcal{M}$-automata [Bri07]. Moreover, it has been shown that it is also a sufficient behavioral relation in order to solve control problems in the framework of timed automata [AMPS98]. However, when considering wider classes of hybrid systems, we will see that this tool is not sufficient anymore for solving control problems in the perfect observation framework.

Definition 3.17. Given a mixed transition system $T = (S, \Gamma, \rightarrow)$, a *time-abstract bisimulation for T* is an equivalence relation $\sim \subseteq S \times S$ such that $\forall q_1, q_1', q_2 \in S$, the two following conditions are satisfied:

$$\forall a \in \Sigma, \left(q_1 \sim q_1' \text{ and } q_1 \xrightarrow{a} q_2\right) \Rightarrow \left(\exists q_2' \in S \text{ s.t. } q_2 \sim q_2' \text{ and } q_1' \xrightarrow{a} q_2'\right)$$

$$\forall \tau \in M^+, \left(q_1 \sim q_1' \text{ and } q_1 \xrightarrow{\tau} q_2\right) \Rightarrow \left(\exists \tau' \in M^+,\ \exists q_2' \in S \text{ s.t. } q_2 \sim q_2' \text{ and } q_1' \xrightarrow{\tau'} q_2'\right)$$

Example 3.18. In this example, we assume a perfect observation framework. Let us consider the $\mathcal{M}$-game $\mathcal{A} = (\mathcal{M}, Q, \mathsf{Goal}, \Sigma, \delta, \gamma)$ where $\mathcal{M} = \langle \mathbb{R}, <, +, 0, 1, \equiv_2 \rangle$ ($\equiv_2$ denotes the "modulo 2" relation), $Q = \{q_1, q_2, q_3\}$, $\mathsf{Goal} = \{q_2\}$, $\Sigma = \Sigma_c \cup \Sigma_u$ where $\Sigma_c = \{c\}$ (resp. $\Sigma_u = \{u\}$) is the set of controllable (resp. uncontrollable) actions. The dynamics in q_1, $\gamma_{q_1} : \mathbb{R}^+ \times \{0,1\} \times \mathbb{R}^+ \rightarrow \mathbb{R}^+ \times \{0,1\}$ is defined as $\gamma_{q_1}(x_1, x_2, t) = (x_1 + t, x_2)$.



(a) The $\mathcal{M}$-game $\mathcal{A}$ (b) Dynamics in q_1

Figure 3: Time-abstract bisimulation does not preserve winning states

We consider the partition depicted on Figure 3(b). The guard g_C is satisfied on C-states and the guard g_B is satisfied on B-states. Note that this partition is compatible with Goal and w.r.t. discrete transitions.

In this game, the controller can win when it enters a C-state by performing action c and it loses when entering a B-state because it cannot prevent the environment from performing a u and going in the losing state q_3.

It follows that the state $s_1 = (q_1, (0, 1))$ is losing, whereas the state $s_2 = (q_1, (0, 0))$ is winning. However, the equivalence relation induced by the partition $\{A, B, C\}$ is a time-abstract bisimulation: the two states s_1 and s_2 are thus time-abstract bisimilar, but not equivalent for the game. It follows that time-abstract bisimulation is not correct for solving control problems, in the sense that a time-abstract bisimulation cannot always distinguish between winning and losing states.

Problem 3.19. Let $\mathcal{M}$ be a structure and $\mathcal{A}$ an $\mathcal{M}$-game. A partition respecting Goal and inducing a time-abstract bisimulation on $Q \times V_2$ does not necessarily respect the set of winning states of $\mathcal{A}$.

4. The Suffix and the Superword Abstractions

In this section we explain how to encode symbolically trajectories of dynamical systems with "words". We will present two different encodings (or abstractions) depending on the observation framework (perfect or partial) we assume.

4.1. **Perfect Observation and the Suffix Abstraction.** In this subsection, we review the word encoding technique introduced in [BMRT04] in order to study o-minimal hybrid systems. We focus on the *suffix partition* introduced in [Bri07]. This encoding will be suitable in order to study control reachability problem in the perfect observation framework (see Subsection 5.3). We first explain how to build words associated with trajectories. Given a dynamical system $(\mathcal{M}, \gamma)$ and a finite partition $\mathcal{P}$ of V_2, given $x \in V_1$ we associate a word with the trajectory $\Gamma_x = \{\gamma(x, t) \mid t \in V\}$ in the following way. We consider the sets $\{t \in V \mid \gamma(x, t) \in P\}$ for $P \in \mathcal{P}$. This gives a partition of the time V. In order to define a word on $\mathcal{P}$ associated with the trajectory determined by x, we need to define the set of intervals $\mathcal{F}_x = \big\{I \mid I$ is a time interval or a point and is maximal for the property "$\exists P \in \mathcal{P},\ \forall t \in I,\ \gamma(x, t) \in P$"$\big\}$. For each x, the set $\mathcal{F}_x$ is totally ordered by the order induced from M. This allows us to define *the word on* $\mathcal{P}$ *associated with the trajectory* Γ_x denoted ω_x.

Definition 4.1. Given $x \in V_1$, *the word associated with* Γ_x is given by the function $\omega_x : \mathcal{F}_x \to \mathcal{P}$ defined by $\omega_x(I) = P$, where $I \in \mathcal{F}_x$ is such that $\forall t \in I$, $\gamma(x, t) \in P$.

The set of words associated with $(\mathcal{M}, \gamma)$ over $\mathcal{P}$ gives in some sense a complete *static* description of the dynamical system $(\mathcal{M}, \gamma)$ through the partition $\mathcal{P}$. In order to recover the *dynamics*, we need further information.

Given a point x of the input space V_1, we have associated with x a trajectory Γ_x and a word ω_x. If we consider (x, t) a point of the space-time $V_1 \times V$, it corresponds to a point $\gamma(x, t)$ lying on Γ_x. To recover in some sense the position of $\gamma(x, t)$ on Γ_x from ω_x, we associate with (x, t) a suffix of the word ω_x denoted $\omega_{(x,t)}$. The construction of $\omega_{(x,t)}$ is similar to the construction of ω_x, we only need to consider the sets of intervals $\mathcal{F}_{(x,t)} = \big\{I \cap \{t' \in V \mid t' \geqslant t\} \mid I \in \mathcal{F}_x\big\}$.

Let us notice that given (x, t) a point of the space-time $V_1 \times V$ there is a unique suffix $\omega_{(x,t)}$ of ω_x associated with (x, t). Given a point $y \in V_2$ it may have several (x, t) such that

$\gamma(x,t) = y$ and so several suffixes are associated with y. In other words, given $y \in V_2$, the *future* of y is non-deterministic, and a single suffix $\omega_{(x,t)}$ is thus not sufficient to recover the dynamics of the transition system through the partition $\mathcal{P}$. To encode the dynamical behavior of a point y of the output space V_2 through the partition $\mathcal{P}$, we introduce the notion of suffix abstraction (called suffix dynamical type in [Bri07, Bri06]) of a point y w.r.t. $\mathcal{P}$.

Definition 4.2. Given a dynamical system $(\mathcal{M}, \gamma)$, a finite partition $\mathcal{P}$ of V_2, a point $y \in V_2$, the *suffix abstraction* of y w.r.t. $\mathcal{P}$ is denoted $\mathsf{Suf}_{\mathcal{P}}(y)$ and defined by $\mathsf{Suf}_{\mathcal{P}}(y) = \{\omega_{(x,t)} \mid \gamma(x,t) = y\}$.

This allows us to define an equivalence relation on V_2. Given y_1, $y_2 \in V_2$, we say that they are *suffix-equivalent* if and only if $\mathsf{Suf}_{\mathcal{P}}(y_1) = \mathsf{Suf}_{\mathcal{P}}(y_2)$. We denote $\mathsf{Suf}\,(\mathcal{P})$ the partition induced by this equivalence, which we call the *suffix partition* w.r.t. $\mathcal{P}$. We say that a partition $\mathcal{P}$ is *suffix-stable* if $\mathsf{Suf}(\mathcal{P}) = \mathcal{P}$ (it implies that if y_1 and y_2 belong to the same piece of $\mathcal{P}$ then $\mathsf{Suf}_{\mathcal{P}}(y_1) = \mathsf{Suf}_{\mathcal{P}}(y_2)$).

To understand the suffix abstraction technique, we provide several examples.

Example 4.3. We start with example 3.10. The suffix abstraction in $(0,0)$ is composed of a unique suffix $ACBC$ because any trajectory leaving $(0,0)$ crosses exactly once the spiral at some point. By looking at Fig. 2 one can convince oneself that the suffixes associated with the other points of the plane are given by suffixes of $ACBC$; for instance, the points lying on the spiral (the piece B) have suffix BC.

Example 4.4. We first consider a two dimensional timed automata dynamics (see Example 3.2). In this case we have that $\gamma(x_1, x_2, t) = (x_1 + t, x_2 + t)$. We associate with this dynamics the partition $\mathcal{P} = \{A, B\}$ where $B = [1,2]^2$ and $A = \mathbb{R}^2 \setminus B$. In this example the suffix partition is made of three pieces, which are depicted in Figure 4.

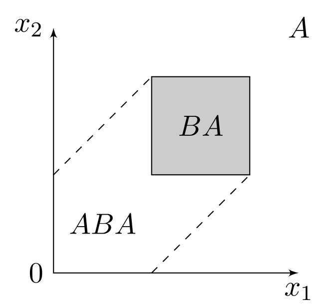
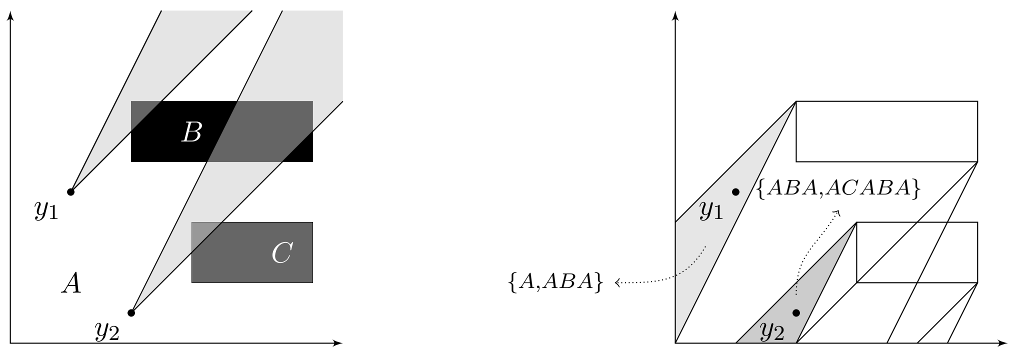


Figure 4: Suffixes for the timed automata dynamics

The suffix abstraction allows to encode more sophisticated continuous dynamics than the previous suffix encoding of a trajectory. In the next example we recover in some sense the continuous dynamics of *rectangular automata* [HKPV98], which requires to use the suffix abstraction (some of the points do not have a unique suffix).

Example 4.5. We consider the dynamical system $(\mathcal{M}, \gamma)$ where $\mathcal{M} = \langle \mathbb{R}, +, \cdot, 0, 1, < \rangle$ and $\gamma : \mathbb{R}^2 \times [1,2] \times \mathbb{R}^+ \to \mathbb{R}^2$ is defined by $\gamma(x_1, x_2, p, t) = (x_1 + t, x_2 + p \cdot t)$. We associate with this dynamical system the partition $\mathcal{P} = \{A, B, C\}$ where $B = [2,5] \times [3,4]$, $C = [3,5] \times [1,2]$

and $A = \mathbb{R}^2 \setminus (B \cup C)$ (see Figure 5(a)). Let us focus on the suffix abstractions of the two points $y_1 = (1, 2.5)$ and $y_2 = (2, 0.5)$. We have that $\mathrm{Suf}_{\mathcal{P}}(y_1) = \{A, ABA\}$ and $\mathrm{Suf}_{\mathcal{P}}(y_2) = \{ABA, ACABA\}$. Though several points have several possible suffixes, the partition induced by the suffix abstraction is finite and illustrated in Figure 5(b).



(a) The dynamics (b) The suffix partition

Figure 5: A rectangular dynamics

4.2. **Partial Observation and the Superword Abstraction.** The suffix-partition proposed in Subsection 4.1 is not suitable for the partial observation framework. We will intuitively convince the reader of this fact. Let $(\mathcal{M}, \gamma)$ be a dynamical system, y be a point of V_2 and $\mathcal{P}$ be a partition of V_2. Since several trajectories cross the point y, there exist several y' such that $y \xrightarrow{\tau} y'$, for some $\tau \in M^+$. In the partial observation framework, the controller does not know which trajectory will be chosen by the environment and have to choose a pair (τ, c) independently. In particular, starting from y, one can potentially be in several different pieces of $\mathcal{P}$ after τ time units. The notion of suffix abstraction is not sufficient in order to capture these behaviors, that is why we now associate a word ω_y on $2^{\mathcal{P}}$ with a given $y \in V_2$. We will see in Subsection 5.2 that this new encoding is suitable in order to study control reachability problem in the partial observation framework. In order to define the word on $2^{\mathcal{P}}$ associated with $y \in V_2$, we need to introduce further definitions.

Definition 4.6. Let y be a point of V_2 and τ be a time in M^+.

$$\mathcal{F}_y(\tau) = \big\{P \in \mathcal{P} \mid \exists x \in M^{k_1}\ \exists t \in M\ \gamma(x,t) = y \text{ and } \gamma(x, t+\tau) \in P\big\}.$$

The set $\mathcal{F}_y(\tau)$ represents the set of pieces that we have potentially reached after τ time units when starting from y.

Definition 4.7. Let y be a point of V_2.

$$\mathcal{F}_y = \big\{I \ \mid\ I \text{ is a time interval and is maximal for the property } \exists S \in 2^{\mathcal{P}}\ \forall \tau \in I\ \ \mathcal{F}_y(\tau) = S\big\}$$

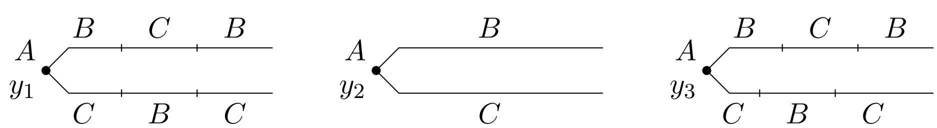
For each $y \in V_2$, the set $\mathcal{F}_y$ exactly consists of the connected components of the sets $\{\tau \in M^+ \mid \mathcal{F}_y(\tau) = S\}$, for $S \in 2^{\mathcal{P}}$. We can now define the superword $\mathsf{Sup}_{\mathcal{P}}(y)$ associated with a given $y \in V_2$.

Definition 4.8. Let $(\mathcal{M},\gamma)$ be a dynamical system, y be a point of V_2, and $\mathcal{P}$ be a partition of V_2. *The superword associated with* y is given by the function $\mathsf{Sup}_{\mathcal{P}}(y) : \mathcal{F}_y \to 2^{\mathcal{P}}$ defined by:

$$\mathsf{Sup}_{\mathcal{P}}(y)(I) = S \qquad \text{where } I \in \mathcal{F}_y \text{ is such that } \forall \tau \in I \;\; \mathcal{F}_y(\tau) = S.$$

Let us notice that given $(\mathcal{M},\gamma)$ a dynamical system, $\mathcal{P}$ a partition of V_2, and y a point of V_2, there exists a unique superword $\mathsf{Sup}_{\mathcal{P}}(y)$ associated with y. If $(\mathcal{M},\gamma)$ is a dynamical system and $\mathcal{P}$ a finite partition of V_2, we write $\mathsf{Sup}(\mathcal{P})$ for the partition induced by superwords. We say that a partition $\mathcal{P}$ is *superword-stable* if $\mathsf{Sup}(\mathcal{P}) = \mathcal{P}$. Let us illustrate this new notion on examples.

Example 4.9. Let us consider the three dynamical systems depicted on Figures 6. In the three cases, the dynamical system consists of two trajectories exiting the point y_i. What differs in the three systems is the way the partition $\mathcal{P} = \{A, B, C\}$ is crossed. We are interested in the superword associated with y_i. For the two first dynamical systems we have that $\mathsf{Sup}_{\mathcal{P}}(y_1) = \mathsf{Sup}_{\mathcal{P}}(y_2) = \{A\}\{B, C\}$, and for the last one we have that $\mathsf{Sup}_{\mathcal{P}}(y_3) = \{A\}\{B, C\}\{B\}\{B, C\}\{C\}\{B, C\}$.



(a) $\{A\}\{B, C\}$ (b) $\{A\}\{B, C\}$ (c) $\{A\}\{B, C\}\{B\}\{B, C\}\{C\}\{B, C\}$

Figure 6: Suffix and superword are not comparable

Let us notice that the notions of *suffix abstraction* and *superword abstraction* are incomparable. To illustrate this fact, let us consider again the three dynamical systems of Figure 6. We have that $\mathsf{Sup}_{\mathcal{P}}(y_1) = \mathsf{Sup}_{\mathcal{P}}(y_2) \neq \mathsf{Sup}_{\mathcal{P}}(y_3)$. Let us now consider the suffix abstractions of these points:

$$\mathsf{Suf}(y_1) = \{ABCB, ACBC\} \;;\; \mathsf{Suf}(y_2) = \{AB, AC\} \;;\; \mathsf{Suf}(y_3) = \{ABCB, ACBC\}.$$

This shows that the superword abstraction can distinguish between y_1 and y_3, but cannot distinguish between y_1 and y_2, although the suffix abstraction can distinguish between y_1 and y_2, but cannot distinguish between y_1 and y_3.

5. Solving an $\mathcal{M}$-Game

In this section we first present a general procedure to compute the set of winning states for an $\mathcal{M}$-game under partial observation. We then show that if a partition is superword-stable, the procedure can be performed symbolically on pieces of the partition. The procedure described is not always effective and we will later point out specific $\mathcal{M}$-structures for which each step of the procedure is computable. By Proposition 3.16, we know that the perfect observation control problem can be seen as a special case of the partial observation framework; however at the end of this section, we explain how the suffix partition can be used in order to directly solve the perfect observation control problem.

5.1. **Controllable Predecessors under Partial Observation.** As for classical reachability games [GTW02], one way of computing winning states is to compute the *attractor* of goal states by iterating a *controllable predecessor* operator. Let $\mathcal{A} = (\mathcal{M}, Q, \mathsf{Goal}, \Sigma, \delta, \gamma)$ be an $\mathcal{M}$-game. For $W \subseteq Q \times V_2$, $a \in \Sigma_c$ and $u \in \Sigma_u$ we first define the notion of controllable discrete predecessors. For every $a \in \Sigma = \Sigma_c \cup \Sigma_u$, we have

$$\mathsf{Pred}_a(W) = \left\{ (q,y) \in Q \times V_2 \;\middle|\; \begin{array}{l} a \text{ is enabled in } (q,y), \\ \text{and } \forall (q',y') \in Q \times V_2, \\ \Big((q,y) \xrightarrow{a} (q',y') \Rightarrow (q',y') \in W \Big) \end{array} \right\}.$$

The intuition of this operator is the following: a state is in $\mathsf{Pred}_a(W)$ if action a can be done from (q,y), and whichever transition is taken leads to a state in W (action a ensures W in one step). We also define $\mathsf{cPred}(W) = \bigcup_{c \in \Sigma_c} \mathsf{Pred}_c(W)$ and $\mathsf{uPred}(W) = \bigcup_{u \in \Sigma_u} \mathsf{Pred}_u(W)$.

As for timed and hybrid games [AMPS98, HHM99], we also define a *safe time predecessor* of a set W w.r.t. a set W', that is specific to the partial observation framework: a state (q,y) is in $\mathsf{time\text{-}Pred}_{\mathsf{partial}}(W,W')$ if a delay τ can be chosen such that for all trajectories starting from (q,y), one can let τ time units pass avoiding W' and then reach $(q',y') \in W$. Formally the operator $\mathsf{time\text{-}Pred}_{\mathsf{partial}}$ is defined as follows:

$$\mathsf{time\text{-}Pred}_{\mathsf{partial}}(W,W') = \left\{ (q,y) \in Q \times V_2 \;\middle|\; \begin{array}{l} \exists \tau \in M^+,\ \forall (x,t) \in V_1 \times V \text{ s.t.} \\ \gamma_q(x,t) = y, \text{ and } (q,y) \xrightarrow{\tau}_{x,t} (q',y') \\ \text{implies } \Big((q',y') \in W \text{ and } \mathsf{Post}^{q,x}_{[t,t+\tau]} \subseteq \overline{W'} \Big) \end{array} \right\}.$$

where $\mathsf{Post}^{q,x}_{[t,t+\tau]} = \{\gamma_q(x,t') \mid t \leqslant t' \leqslant t+\tau\}$.

The *controllable predecessor* operator under partial observation π_{partial} is then defined as:

$$\pi_{\mathsf{partial}}(W) = W \cup \bigcup_{a \in \Sigma_c} \mathsf{time\text{-}Pred}_{\mathsf{partial}}(\mathsf{Pred}_a(W), \mathsf{uPred}(\overline{W})).$$

Remark 5.1. Note that the operator π_{partial} is definable in any expansion of an ordered group. Hence, if W is definable, so is $\pi_{\mathsf{partial}}(W)$.

Example 5.2. We first illustrate the computation of the operator π_{partial} on Example 3.10 (see page 8). In this case, π_{partial} does not induce a winning strategy from $(q_1,(0,0))$ under partial observation. Setting $W = \mathsf{Goal} \times V_2 = \{q_2\} \times V_2$, we have that $\pi_{\mathsf{partial}}(W)$ does not contain the point $(q_1,(0,0))$ because there is no uniform choice for a positive delay τ before taking action c so that the spiral (area B) can be avoided. Notice however that $\pi_{\mathsf{partial}}(W)$ is not empty because it includes all points different from $(q_1,(0,0))$ (from which there is a unique trajectory).

Remark 5.3. Note also that due to the partial observation assumption, in the definition of π_{partial}, the action a for controlling the system has to be chosen before choosing the delay τ. Indeed, the controller does not know which precise trajectory will be chosen by the environment, in particular, action a should be available after time τ independently of the choice of trajectory made by the environment. This is illustrated in the next example.

Example 5.4. Let us consider the $\mathcal{M}$-game $\mathcal{A}$ depicted on Figure 7(a) where $\mathsf{Goal} = \{q_2, q_3\}$ and where $c_1, c_2 \in \Sigma_c$ are distinct controllable actions. The dynamics in q_1 is depicted on Figure 7(b), roughly speaking, it consists of of two trajectories exiting the point y. perfect observation from y; indeed depending on the trajectory we are following, we will either play (τ, c_1) or (τ, c_2), for some well-chosen $\tau \in \mathbb{R}^+$. However, there is no winning strategy under partial observation from y. Although we can find $\tau \in \mathbb{R}^+$ such that a controllable action will be (safely) available (from y) after τ time units, we are unable to tell which controllable action will be taken.

In fact if $W = \mathsf{Goal} \times V_2$ we have that $\pi_{\mathsf{partial}}(W) = \{(q_1, z) \mid z \in V_2 \backslash \{y\}\}$. Indeed if $(q_1, z) \neq (q_1, y)$, the controller can deduce the trajectory from the current state and choose its action accordingly.

(a) The $\mathcal{M}$-game $\mathcal{A}$ (b) Dynamics in q_1

Figure 7:

The next proposition states the soundness of this operator for computing winning states in the games under a partial observation hypothesis.

Problem 5.5. Let $\mathcal{A} = (\mathcal{M}, Q, \mathsf{Goal}, \Sigma, \delta, \gamma)$ be an $\mathcal{M}$-game. If there exists $n \in \mathbb{N}$ s.t. $\pi^n_{\mathsf{partial}}(\mathsf{Goal}) = \pi^{n+1}_{\mathsf{partial}}(\mathsf{Goal})$ then $\pi^*_{\mathsf{partial}}(\mathsf{Goal}) = \pi^n_{\mathsf{partial}}(\mathsf{Goal})$ is the set of winning states of $\mathcal{A}$ under partial observation.

Proof. We first prove that if $(q, y) \in \pi^*_{\mathsf{partial}}(\mathsf{Goal})$ then there exists a winning strategy under partial observation from (q, y). To this aim, we define a memoryless winning strategy from any $(q, y) \in \pi^*_{\mathsf{partial}}(\mathsf{Goal})$. By notation misuse, we define the strategy λ on states (q, y) instead of executions.

We define a strategy λ on all sets $\bigcup_{0 \leqslant i \leqslant k} \pi^i_{\mathsf{partial}}(\mathsf{Goal})$ by induction on k, and prove that it is a winning strategy. If $k = 0$, we assume λ is defined nowhere, it is thus winning from all states in Goal.

Suppose now that λ is already defined on $W = \bigcup_{0 \leqslant i \leqslant k} \pi^i_{\mathsf{partial}}(\mathsf{Goal})$ and is winning on these states. We now define λ on $\pi_{\mathsf{partial}}(W)$. Let $(q, y) \in Q \times V_2$: if $(q, y) \in W$, λ is already defined; if $(q, y) \in \pi_{\mathsf{partial}}(W) \setminus W$, then we know that there exists $a \in \Sigma_c$ with $(q, y) \in \mathsf{time\text{-}Pred}_{\mathsf{partial}}\big(\mathsf{Pred}_a(W), \mathsf{uPred}(\overline{W})\big)$. There exists $\tau \in M^+$ with (τ, a) enabled[5] in (q, y) such that for every (x, t) if $\gamma_q(x, t) = y$, then $(q, y) \xrightarrow{\tau, a}_{x,t} (q', y')$, $(q', y') \in W$ and $\mathsf{Post}^{q,x}_{[t,t+\tau]} \subseteq \overline{\mathsf{uPred}(\overline{W})}$. We set $\lambda(q, y) = (\tau, a)$ and show that this is a winning choice.

We show by induction on k that λ is winning for each state of $W = \bigcup_{0 \leqslant i \leqslant k} \pi^i_{\mathsf{partial}}(\mathsf{Goal})$. This is immediate for $k = 0$. Suppose now that the result is true for k and let $(q, y) \in \pi_{\mathsf{partial}}(W)$. Let $\rho = (q, x, t, y) \xrightarrow{\tau_1, a_1} (q_1, x_1, t_1, y_1) \xrightarrow{\tau_2, a_2} \ldots$ be an execution compatible with λ. We have that either $\tau_1 = \tau$ and $a_1 = a$, in which case $(q_1, y_1) \in W$, or $\tau_1 \leqslant \tau$ and $a_1 \in \Sigma_u$, in which case $(q, y) \xrightarrow{\tau_1}_{x,t} (q', y') \xrightarrow{a_1} (q_1, y_1)$ with $(q', y') \notin \mathsf{uPred}(\overline{W})$ so $(q_1, y_1) \in W$. In both cases, $(q_1, y_1) \in W$ so by induction hypothesis, ρ is winning.

[5]We say that $(\tau, a) \in M^+ \times \Sigma$ is enabled in (q, y) if there exists $(x, t) \in V_1 \times V$ such that $\gamma(x, t) = y$ and (τ, a) is enabled in (q, x, t, y).

We now show that if there exists a strategy under partial observation λ winning from (q,y) then $(q,y) \in \pi^*_{\mathsf{partial}}(\mathsf{Goal})$. Set $W = \pi^*_{\mathsf{partial}}(\mathsf{Goal})$, by contradiction suppose that $(q,y) \notin W$, we will construct a non-winning execution compatible with λ. By hypothesis $\pi_{\mathsf{partial}}(W) = W$ so $(q,y) \notin \pi_{\mathsf{partial}}(W)$, it follows that for all $a \in \Sigma_c$, for all $\tau \in M^+$ there exists $(x,t) \in V_1 \times V$ such that $\gamma_q(x,t) = y$, and $(q,y) \to^{\tau}_{x,t} (q',y')$ implies $(q',y') \notin \mathsf{Pred}_a(W)$ or $\mathsf{Post}^{q,x}_{[t,t+\tau]} \cap \mathsf{uPred}(\overline{W}) \neq \varnothing$. Let $(\tau,a) = \lambda(q,y)$ (as λ is a strategy under partial observation it does not depend of x and t) and let $(x,t) \in V_1 \times M^+$ be as in the previous statement.

There exists (q_1,x_1,t_1,y_1) with $(q_1,y_1) \notin W$ such that either $(q,x,t,y) \xrightarrow{\tau,a} (q_1,x_1,t_1,y_1)$ or there exists $\tau' \leqslant \tau$ and $u \in \Sigma_u$ with $(q,x,t,y) \xrightarrow{\tau',u} (q_1,x_1,t_1,y_1)$. In both cases, the constructed execution is compatible with λ. As $(q_1,y_1) \notin W$ we can repeat the same argument and construct inductively an execution $\rho = (q,x,t,y) \xrightarrow{\tau_1,a_1} (q_1,x_1,t_1,y_1) \xrightarrow{\tau_2,a_2} \ldots$ compatible with λ and such that for every i, $(q_i,x_i,t_i,y_i) \notin W$. By definition of W, for every i, $q_i \notin \mathsf{Goal}$, which contradicts the assumption that λ is a winning strategy. □

$\pi^*_{\mathsf{partial}}(\mathsf{Goal})$, but this does not imply that we can compute this set, as some $\mathcal{M}$-structures have an undecidable theory. The following corollary states that if some conditions on the structure and on π_{partial} are satisfied, then this procedure provides an algorithmic solution to the control problem.

Corollary 5.6. *Let $\mathcal{M}$ be a structure such that $\mathsf{Th}(\mathcal{M})$ is decidable.*[6] *Let $\mathcal{C}$ be a class of $\mathcal{M}$-games such that for every $\mathcal{A}$ in $\mathcal{C}$, there exists a finite partition $\mathcal{P}$ of $Q \times V_2$ definable in $\mathcal{M}$, respecting* Goal[7]*, and stable under π_{partial}.*[8] *Then the control problem under partial observation in the class $\mathcal{C}$ is decidable. Moreover if $\mathcal{A} \in \mathcal{C}$, the set of winning states under partial observation of $\mathcal{A}$ is computable.*

Proof. Let $\mathcal{M}$ be a structure and $\mathcal{C}$ a class of automata satisfying the hypotheses and take $\mathcal{A} \in \mathcal{C}$. As $\mathcal{P}$ is stable under π_{partial}, $\pi^*_{\mathsf{partial}}(\mathsf{Goal})$ is a finite union of pieces of $\mathcal{P}$. Hence there exists $n \in \mathbb{N}$ such that $\pi^*_{\mathsf{partial}}(\mathsf{Goal}) = \pi^n_{\mathsf{partial}}(\mathsf{Goal})$. Thus proposition 5.5 shows that the set of winning states is $\pi^*_{\mathsf{partial}}(\mathsf{Goal})$.

As π_{partial} and Goal are definable, we have that $\pi^i_{\mathsf{partial}}(\mathsf{Goal})$ is definable and as $\mathsf{Th}(\mathcal{M})$ is decidable we can test if $\pi^i_{\mathsf{partial}}(\mathsf{Goal}) = \pi^{i+1}_{\mathsf{partial}}(\mathsf{Goal})$, we can thus effectively find a representation of $\pi^*_{\mathsf{partial}}(\mathsf{Goal})$.

As $\mathsf{Th}(\mathcal{M})$ is decidable, if a state (q,y) is definable we can test if $(q,y) \in \pi^*_{\mathsf{partial}}(\mathsf{Goal})$. It follows that the control problem in an $\mathcal{M}$-structure is decidable. □

5.2. **Superwords and the π_{partial} Operator.** We now present a sufficient condition for a partition to be stable under the operator π_{partial}: we require that the partition is stable under Pred_a (for all $a \in \Sigma$) to handle the discrete part of the automaton and we show that the stability by superwords is fine enough to be correct for solving control problems under partial observation.

[6]We recall that a theory $\mathsf{Th}(\mathcal{M})$ is decidable iff there is an algorithm which can determine whether or not any sentence (*i.e.*, a formula with no free variable.) is a member of the theory (*i.e.*, is true). We suggest to readers interested in general decidability issues on o-minimal hybrid systems to refer to Section 5 of [BM05].

[7]*I.e.*, Goal is a union of pieces of $\mathcal{P}$.

[8]Meaning that if P is a piece of $\mathcal{P}$ then $\pi_{\mathsf{partial}}(P)$ is a union of pieces of $\mathcal{P}$.

Problem 5.7. Let $\mathcal{A}$ be an $\mathcal{M}$-game and $\mathcal{P}$ be a partition of $Q \times V_2$. If $\mathcal{P}$ respects Goal, is stable under Pred_a (for all $a \in \Sigma$) and superword-stable, then $\mathcal{P}$ is stable under the operator π_{partial}.

Proof. We fix a location q of the automaton and we take $y_1, y_2 \in V_2$ such that there exists $A \in \mathcal{P}$ with $y_1, y_2 \in A$. We now show that if $y_1 \in \pi_{\mathsf{partial}}(X)$, for some $X \in \mathcal{P}$ then $y_2 \in \pi_{\mathsf{partial}}(X)$. In case $y_1 \in X$ then $X = A$ and $y_2 \in Y$.

We assume $y_1 \in \pi_{\mathsf{partial}}(X) \setminus X$. There exists $a \in \Sigma_c$ and $\tau_1 \in M^+$ such that for all $(x,t) \in V_1 \times V$ with $\gamma_q(x,t) = y_1$ and for all y_1' such that $y_1 \xrightarrow{\tau_1}_{x,t} y_1'$, we have that $y_1' \in \mathsf{Pred}_a(X)$, and $\mathsf{Post}^{q,x}_{[t,t+\tau_1]} \subseteq \overline{\mathsf{uPred}(\overline{X})}$. Let us now express the previous condition in term of superword. Assume that

$$\mathsf{Sup}_{\mathcal{P}}(y_1) = S_1 S_2 \cdots S_k, \quad \text{where } S_i \in 2^{\mathcal{P}},$$

the previous condition means that $\mathsf{Sup}_{\mathcal{P}}(y_1)$ contains a prefix $S_1 \cdots S_l$ is such that:

- for all $P_i \in S_l$, we have that $P_i \subseteq \mathsf{Pred}_a(X)$ (this condition makes sense since $\mathcal{P}$ is stable under Pred_a; indeed, *a priori* we only have that there exists $y_1' \in P_i$ such that $y_1' \in \mathsf{Pred}_a(X)$, the stability of $\mathcal{P}$ under Pred_a implies that $P_i \subseteq \mathsf{Pred}_a(X)$),
- for all $j \leqslant l$, for all $P_i \in S_j$, we have that $\mathsf{uPred}(\overline{X}) \cap P_i = \varnothing$ (again this condition makes sense since $\mathcal{P}$ is stable under Pred_a).

Since $\mathcal{P} = \mathsf{Sup}(\mathcal{P})$ and both y_1 and y_2 belong to the same piece of $\mathcal{P}$, we have that $\mathsf{Sup}_{\mathcal{P}}(y_1) = \mathsf{Sup}_{\mathcal{P}}(y_2) = S_1 S_2 \cdots S_k$. In particular, we can find $\tau_2 \in M^+$ such that if $y_2 \xrightarrow{\tau_2} y_2'$, we have that y_2' corresponds to the letter S_l. Thus we have that $y_2' \in \mathsf{Pred}_a(X)$ and $\mathsf{Post}^{q,x}_{[t,t+\tau_2]} \subseteq \overline{\mathsf{uPred}(\overline{X})}$, *i.e.* $y_2 \in \pi_{\mathsf{partial}}(X)$. □

As an immediate corollary of this proposition and of Corollary 5.6, we get the following general decidability result.

Corollary 5.8. *Let $\mathcal{M}$ be a structure such that* $\mathsf{Th}(\mathcal{M})$ *is decidable. Let $\mathcal{C}$ be a class of $\mathcal{M}$-games such that for every $\mathcal{A}$ in $\mathcal{C}$, there exists a finite partition $\mathcal{P}$ of $Q \times V_2$ definable in $\mathcal{M}$, respecting* Goal*, superword-stable, and stable under* Pred_a *for every action $a \in \Sigma$. Then the control problem under partial observation (Problem 3.13) in the class $\mathcal{C}$ is decidable, and if $\mathcal{A} \in \mathcal{C}$, the set of winning states under partial observation of $\mathcal{A}$ is computable.*

5.3. **A Note on the Perfect Observation Framework.** We briefly discuss the perfect observation framework. We have already seen that it is a special case of the partial observation framework (see Proposition 3.16). Hence, we can reuse the previous results and get decidability and computability results. However, we can also define an appropriate controllable predecessor operator π_{perfect} that will be correct in the perfect observation framework. The new operator π_{perfect} is just a twist of the previous operator, which we define as:

$$\pi_{\mathsf{perfect}}(W) = W \cup \mathsf{time\text{-}Pred}_{\mathsf{perfect}}\left(\mathsf{cPred}(W), \mathsf{uPred}(\overline{W})\right)$$

where $\mathsf{time\text{-}Pred}_{\mathsf{perfect}}$ existentially quantifies on pairs (x,t) such that $y = \gamma_q(x,t)$ (instead of universally quantifying on those pairs, as in $\mathsf{time\text{-}Pred}_{\mathsf{partial}}$).

Remark 5.9. In the perfect observation framework, the controller is aware of the precise trajectory that will be followed, hence his choice of action can be done after his choice of delay contrarily to the partial observation case (remember Remark 5.3). That is why the union over actions is put within the scope of the safe time predecessor in π_{perfect}.

Applying similar reasoning as in the previous sections, we can prove that $\pi^*_{\mathsf{perfect}}(\mathsf{Goal})$ corresponds to the set of winning states of $\mathcal{A}$, and that a partition, which is both stable under Pred_a (for every $a \in \Sigma$) and suffix-stable, is actually correct for solving control problems in the perfect observation framework. We can thus state the following theorem.

Theorem 5.10. *Let $\mathcal{M}$ be a structure such that $\mathsf{Th}(\mathcal{M})$ is decidable. Let $\mathcal{C}$ be a class of $\mathcal{M}$-games such that for every $\mathcal{A}$ in $\mathcal{C}$, there exists a finite partition $\mathcal{P}$ of $Q \times V_2$ definable in $\mathcal{M}$, respecting Goal, suffix-stable, and stable under Pred_a for every action $a \in \Sigma$. Then the control problem under perfect observation (Problem 3.8) in the class $\mathcal{C}$ is decidable, and if $\mathcal{A} \in \mathcal{C}$, the set of winning states under perfect observation of $\mathcal{A}$ is computable.*

Note that being suffix-stable is a stronger condition than being a time-abstract bisimulation [Bri07], and we see here that this is one of the right tools to solve control problems. For instance in Example 3.18 the partition $\mathcal{P}$ is a time-abstract bisimulation but is not suffix-stable. Indeed $s_1, s_2 \in A$ but $\mathrm{Suf}_{\mathcal{P}}(s_1) \neq \mathrm{Suf}_{\mathcal{P}}(s_2)$.

Remark 5.11. Using the results of this section, we recover the results of [AMPS98] about control of timed automata. Note that for the timed automata dynamics (remember Example 3.2) partial or perfect observation do not make a difference (the dynamics is deterministic). Indeed we consider the classical finite partition of timed automata that induces the region graph (see [AD94]). Let us call $\mathcal{P}_R$ this partition, and notice that $\mathcal{P}_R$ is definable in $\langle \mathbb{R}, <, +, 0, 1 \rangle$. $\mathcal{P}_R$ is stable under the action of Pred_a for every action $a \in \Sigma$. By Example 3.2 the continuous dynamics of timed automata is definable in $\langle \mathbb{R}, <, +, 0, 1 \rangle$. Hence it makes sense to encode continuous trajectories of timed automata as words. Then one can easily verify that $\mathsf{Suf}(\mathcal{P}_R) = \mathcal{P}_R$. By Theorem 5.10 we get the decidability and computability of winning states under perfect information in timed games [AMPS98] as a side result.

Corollary 5.12. *The control problem under perfect information in the class of timed automata is decidable. Moreover the set of winning states under perfect observation is computable.*

6. O-Minimal Games

In this section, we focus on the particular case of o-minimal games (*i.e.*, $\mathcal{M}$-games where $\mathcal{M}$ is an o-minimal structure and in which extra assumptions are made on the resets). We first briefly recall definitions and results related to o-minimality [PS86]. We show that existence of finite partitions which are stable w.r.t. the controllable predecessor operator can be guaranteed for o-minimal games. More precisely, we first show that, in this framework, a partition stable under the controllable predecessor operator can easily be obtained *via* the superword abstraction (this is due to the assumptions on the resets). Then, we use properties of o-minimality to prove the finiteness of the previously obtained partition. Finally we focus on o-minimal structures with a decidable theory in order to obtain full decidability and computability results. As in the previous section, we mostly focus on the partial observation framework, but also mention results in the perfect observation framework.

6.1. **O-Minimality.** We recall here the definition of o-minimality and the "*Uniform Finiteness Theorem*" that will be applied later in this section. The reader interested in o-minimality should refer to [vdD98] for further results and an extensive bibliography on this subject.

Definition 6.1. An extension of an ordered structure $\mathcal{M} = \langle M, <, \ldots \rangle$ is *o-minimal* if every definable subset of M is a finite union of points and open intervals (possibly unbounded).

In other words the definable subsets of M are the simplest possible: the ones which are definable in $\langle M, < \rangle$. This assumption implies that definable subsets of M^n (in the sense of $\mathcal{M}$) admit very nice structure theorems (like the *cell decomposition* [KPS86]) or Theorem 6.2 below. The following are examples of o-minimal structures: the ordered group of rationals $\langle \mathbb{Q}, <, +, 0, 1 \rangle$, the ordered field of reals $\langle \mathbb{R}, <, +, \cdot, 0, 1 \rangle$, the field of reals with exponential function, the field of reals expanded by restricted pfaffian functions and the exponential function, and many more interesting structures (see [vdD98, Wil96]). An example of non o-minimal structure is given by $\langle \mathbb{R}, <, \sin, 0 \rangle$, since the definable set $\{x \mid \sin(x) = 0\}$ is not a finite union of points and open intervals. However, let us mention that the structure[9] $\langle \mathbb{R}, +, \cdot, 0, 1, <, \sin_{|[0,2\pi]}, \cos_{|[0,2\pi]} \rangle$ is o-minimal (see [vdD96]).

Theorem 6.2 (Uniform Finiteness [KPS86])**.** *Let $\mathcal{M} = \langle M, <, \ldots \rangle$ be an o-minimal structure. Let $S \subseteq M^m \times M^n$ be definable (in $\mathcal{M}$), we denote by S_a the fiber $\{y \in M^n | (a, y) \in S\}$. Then there is a number $N_S \in \mathbb{N}$ such that for each $a \in M^m$ the set $S_a \subseteq M^n$ has at most N_S definably connected components.*

6.2. **Generalities on O-Minimal Games.**

Definition 6.3. Given $\mathcal{A}$ an $\mathcal{M}$-game, we say that $\mathcal{A}$ is an *o-minimal game* if the structure $\mathcal{M}$ is o-minimal and if all transitions (q, g, a, R, q') of $\mathcal{A}$ belong to[10] $Q \times 2^{V_2} \times \Sigma \times 2^{V_2} \times Q$.

Let us notice that the previous definition implies that given $\mathcal{A}$ an o-minimal game, the guards, the resets and the dynamics are definable in the underlying o-minimal structure. We denote by $\mathcal{P}_{\mathcal{A}}$ the coarsest partition of the state space $S = Q \times V_2$ which respects Goal, and all guards and resets in $\mathcal{A}$. Note that $\mathcal{P}_{\mathcal{A}}$ is a finite definable partition of S.

Due to the strong reset condition we have that $\mathcal{P}_{\mathcal{A}}$ is stable under the action of Pred_a for every action a. This holds by the same argument that allows to decouple the continuous and discrete components of a hybrid system in [LPS00]. Let us also notice that, in the framework of o-minimal games, any refinement of $\mathcal{P}_{\mathcal{A}}$ is stable under the action of Pred_a for every $a \in \Sigma$.

Example 6.4. The continuous dynamics of timed automata (see Example 4.4) is definable in the o-minimal structure $\langle \mathbb{R}, +, 0, 1, < \rangle$. The continuous dynamics of rectangular automata (see Example 4.5) is definable in the o-minimal structure $\langle \mathbb{R}, +, \cdot, 0, 1, < \rangle$. Hence games on timed (resp. rectangular) automata with strong resets are particular cases of o-minimal games. The $\mathcal{M}$-game of Example 3.10 is in fact an o-minimal game; indeed one can see that it can be defined in the structure $\langle \mathbb{R}, +, \cdot, 0, 1, <, \sin_{|[0,2\pi]}, \cos_{|[0,2\pi]} \rangle$ which is o-minimal (see [vdD96]).

[9] $\sin_{|[0,2\pi]}$ and $\cos_{|[0,2\pi]}$ correspond to the sinus and cosinus functions restricted to the segment $[0, 2\pi]$.
[10] This is a particular case of reset for $\mathcal{M}$-game where we consider only constant functions for resets.

6.3. **Solving O-Minimal Games.** In this subsection, we will see how we can (easily) build a partition which is stable under the actions of the controllable predecessor operator. The key ingredients to build this partition will be (i) the strong resets conditions and (ii) the superword abstraction. The finiteness of the obtained partition will be discussed in Subsection 6.4.

Problem 6.5. Let $\mathcal{A}$ be an o-minimal game, and $\mathcal{P}_{\mathcal{A}}$ the partition corresponding to its guards and resets. The superword (resp. suffix) partition $\mathsf{Sup}(\mathcal{P}_{\mathcal{A}})$ (resp. $\mathsf{Suf}(\mathcal{P}_{\mathcal{A}})$) is stable under the action of π_{partial} (resp. π_{perfect}).

Proof. This proposition is not a corollary of Proposition 5.7, as $\mathsf{Sup}(\mathcal{P}_{\mathcal{A}})$ *is not* superword-stable. However, the proof of Proposition 5.7 only relied on the fact that in a superword-stable partition, two points in a piece of the partition have the same superword abstraction, which is precisely what we have in the current case. Hence the previous proof can be mimicked, and we do not write all details. It is worth noting also that we do not use all properties of o-minimal games, but only the strong reset property, which ensures that the partition is stable under Pred_a for every action $a \in \Sigma$. □

6.4. **Definability and Finiteness Issues.** In the previous subsection, we have proved that, given $\mathcal{A}$ an o-minimal game, the partition $\mathsf{Sup}(\mathcal{P}_{\mathcal{A}})$ (resp. $\mathsf{Suf}(\mathcal{P}_{\mathcal{A}})$) is stable under the action of the controllable predecessor operator under the partial (resp. perfect) observation framework. We will now show that this partition is finite. For this we will exploit the finiteness property of o-minimality and in order to do so, we first need to prove that our encodings are definable.

6.4.1. *Definability.* Let $(\mathcal{M},\gamma)$ be a dynamical system and $\mathcal{P}$ be a finite partition of V_2. We now would like to show that in the case of *o-minimal dynamical system* the superword encoding previously discussed can be done in a *definable* way. The approach closely follows the one used in [Bri06, Section 12.2] for the suffix abstraction (called suffix dynamical type in this paper).

Let $(\mathcal{M},\gamma)$ be an o-minimal dynamical system and $\mathcal{P}$ be a finite definable partition of V_2. First let us notice that, since $\mathcal{P}$ is finite and definable, given $S \in 2^{\mathcal{P}}$ one can easily write a first-order formula $\varphi(y,\tau)$ which is true if and only if $\mathcal{F}_y(\tau) = S$ (where $\mathcal{F}_y$ is defined similarly to $\mathcal{F}_x$ – see page 11). Let us give this formula, assuming that $S = \{A_1,\dots,A_n\}$:

$$\varphi_S(y,\tau) \equiv \exists x_1\, \exists t_1\, \cdots\, \exists x_n\, \exists t_n \bigwedge_{i=1,\dots,n} \big(\gamma(x_i,t_i) = y \,\wedge\, \gamma(x_i,t_i+\tau) \in A_i\big)$$
$$\wedge\, \forall x\, \forall t\, \big(\gamma(x,t) = y\big) \Rightarrow \big(\gamma(x,t+\tau) \in A_1 \cup \cdots \cup A_n\big).$$

Thus, for each $y \in V_2$, the set $\mathcal{F}_y$ exactly consists of the connected components of the sets $\{\tau \in M^+ \mid \varphi_S(y,\tau)\}$, for $S \in 2^{\mathcal{P}}$; *i.e.* $\mathcal{F}_y$ is a set of intervals. In order to show that $\mathcal{F}_y$ is first-order definable we need to encode each interval $I \subseteq M$ as a point in some cartesian power of M. An interval $I \subseteq M$ is entirely characterized by *(i)* its end-points and *(ii)* the fact of being right (resp. left) open or closed. For *(i)* we formally need a couple to represent a single end point in order to recover $-\infty$ and $+\infty$ (as in the projective line case). For *(ii)* we can use a binary encoding, let us say 0 means open and 1 closed. Thus any interval $I \subseteq M$ will be encoded by an element $(a_1,a_2,a_3,b_1,b_2,b_3) \in M^6$. For instance, the interval $I = \{x \in \mathbb{R} \mid x \geqslant 5\}$ is encoded by $(5,1,1,1,0,0)$. Thanks to this "trick", one can find a first-order formula φ_y defining $\mathcal{F}_y$. The writing of the formula φ_y is not difficult but rather tedious: different cases have to be considered (depending on whether the interval I, encoded by an element of M^6, is left (resp. right) bounded and left (resp. right) open or closed). Further details of the construction of the formula can be found in [Bri06, Section 12.2].

6.4.2. *Finiteness.* We will now prove that when considering o-minimal dynamical systems, only finitely many finite superwords are needed to encode all possible trajectories.

Problem 6.6. Let $(\mathcal{M}, \gamma)$ be an o-minimal dynamical system and $\mathcal{P}$ be a finite definable partition of V_2. There exists finitely many finite superwords associated with $(\mathcal{M}, \gamma)$ w.r.t. $\mathcal{P}$.

Proof. Given $S \in 2^{\mathcal{P}}$ let us first consider the set

$$\mathcal{F}_y(S) = \{\tau \in M^+ \mid \mathcal{F}_y(\tau) = S\} = \{\tau \in M^+ \mid \varphi_S(y, \tau)\}.$$

By the above discussion, the set $\mathcal{F}_y(S)$ is a definable subset of M. Hence by o-minimality it is a finite union of points and open intervals, in particular, it has only finitely many connected components. By definition of $\mathcal{F}_y$ we have the following equality.

$$|\mathcal{F}_y| = \sum_{S \in 2^{\mathcal{P}}} \Big(\text{number of connected components of } \mathcal{F}_y(S)\Big).$$

Since $\mathcal{P}$ is finite we can conclude that $\mathcal{F}_y$ is finite.

Using the uniform finiteness theorem (Theorem 6.2) we obtain that there exists $N \in \mathbb{N}$ such that for all $y \in V_2$ we have that $\left|\mathcal{F}_y\right| \leqslant N$.

In terms of word encoding, this means that there are only finitely many superwords associated with the points of the (output) space V_2. More precisely, the superwords $\mathsf{Sup}_{\mathcal{P}}(y)$ have lengths uniformly bounded by N. Since the superwords $\mathsf{Sup}_{\mathcal{P}}(y)$ are words on the *finite* alphabet $2^{\mathcal{P}}$, this completes the proof. □

The previous proposition directly implies the finiteness of the partition $\mathsf{Sup}(\mathcal{P})$. Moreover we have that this partition is definable, as stated in the following proposition.

Problem 6.7. Let $(\mathcal{M}, \gamma)$ be an o-minimal dynamical system, $\mathcal{P}$ be a finite definable partition of the output space V_2. The partition $\mathsf{Sup}(\mathcal{P})$ is finite and definable.

Proof. Since there are only finitely many superwords, it suffices to show that given $y \in V_2$ and SW a superword on $\mathcal{P}$ (i.e. a word on $2^{\mathcal{P}}$), we can define (by a first-order formula) that $SW = \mathsf{Sup}_{\mathcal{P}}(y)$. Suppose that $SW = S_1 \cdots S_k \cdots S_n$, where $S_k \in 2^{\mathcal{P}}$. We have that $SW = \mathsf{Sup}_{\mathcal{P}}(y)$ if and only if the following formula holds.

$$\exists \tau_1 \in M^+,\ \exists \tau_2 \in M^+,\ \cdots\ \exists \tau_n \in M^+,\ \exists I_1 \in \mathcal{F}_y,\ I_2 \in \mathcal{F}_y,\ \cdots\ \exists I_n \in \mathcal{F}_y$$

$$(\tau_1 < \tau_2 < \cdots < \tau_n)\ \wedge\ \bigwedge_{k=1}^{n} \mathcal{F}_y(\tau_k) = S_k\ \wedge\ \mathcal{F}_y = \{I_1, I_2, \ldots, I_n\}.$$

Notice that the above formula is first-order since $\mathcal{F}_y$ is first-order definable and testing whether $\mathcal{F}_y(\tau_k) = S_k$ is also first-order definable. □

6.5. **Synthesis of Winning Strategies.** We now prove that given $\mathcal{A}$ an o-minimal game definable in $\mathcal{M}$, we can construct a *definable* strategy (in the same structure $\mathcal{M}$) for the winning states under partial observation. The effectiveness of this construction will be discussed later.

Theorem 6.8. *Given $\mathcal{A}$ an o-minimal game, there exists a definable memoryless winning strategy under partial (resp. perfect) observation for each $(q,y) \in \pi^*_{\mathsf{partial}}(\mathsf{Goal})$ (resp. $\pi^*_{\mathsf{perfect}}(\mathsf{Goal})$).*

Proof. By Proposition 6.5, the partition $\mathsf{Sup}(\mathcal{P}_\mathcal{A})$ is finite, definable and stable under π_{partial}. In particular, there exists thus $n \in \mathbb{N}$ such that $\pi^*_{\mathsf{partial}}(\mathsf{Goal}) = \pi^n_{\mathsf{partial}}(\mathsf{Goal})$. Hence, by Proposition 5.5, $\pi^n_{\mathsf{partial}}(\mathsf{Goal})$ is the set of winning states.

Given $(q,y) \in \pi^n_{\mathsf{partial}}(\mathsf{Goal})$, we know that there exists a winning strategy from (q,y). We now have to point out a definable winning strategy from (q,y). Following the proof of Proposition 5.5, we build the definable strategy by induction on the number of iterations of π_{partial}. Let us suppose we have already built a strategy on each piece of $W = \bigcup_{0 \leqslant i \leqslant k} \pi^i_{\mathsf{partial}}(\mathsf{Goal})$, let us now consider $\pi_{\mathsf{partial}}(W) \setminus W$.

By Proposition 6.5, we know that $\pi_{\mathsf{partial}}(W) \setminus W$ is a finite union of pieces of $\mathsf{Sup}(\mathcal{P}_\mathcal{A})$. Let P be one of these pieces. We know that P corresponds to a finite superword on $\mathcal{P}_\mathcal{A}$. Thus given $(q,y) \in P$ we have that

$$\mathsf{Sup}_{\mathcal{P}_\mathcal{A}}(y) = S_1 S_2 \cdots S_k, \qquad \text{where } S_i \in 2^{\mathcal{P}_\mathcal{A}}.$$

Since $(q,y) \in \pi_{\mathsf{partial}}(W) \setminus W$, the superword $\mathsf{Sup}_{\mathcal{P}_\mathcal{A}}(y)$ contains a prefix $S_1 \cdots S_l$ such that there is $a \in \Sigma_c$ with:

- for all $P_i \in S_l$, $P_i \subseteq \mathsf{Pred}_a(W)$,
- for all $j \leqslant l$, for all $P_i \in S_j$, $\mathsf{uPred}(\overline{W}) \cap P_i = \varnothing$.

Since for all $P_i \in S_l$, we have that $P_i \subseteq \mathsf{Pred}_a(W)$, the controllable action $a \in \Sigma_c$ is such that given any $(q,y) \in S_l$ a transition labelled by a is enabled and all such transitions lead to W. The strategy for (q,y) will be to perform action a after some delay. We now explain how to choose this delay.

Let (q,y) be such that $(q,y) \in P$. Let us consider $\mathsf{Time}(y)$ the subset of M^+ defined as follows:

$$\mathsf{Time}(y) = \{\tau \in M^+ \mid \exists y' \in S_l \text{ such that } (q,y) \xrightarrow{\tau} (q,y')\}.$$

This set is definable since S_l is definable.

By o-minimality, we have that $\mathsf{Time}(y)$ is a finite union of points and open intervals. Let us denote by I the leftmost point or interval. Let us notice that I is definable. If I has a minimum m, we define $\lambda(q,y) = (m,c)$. Otherwise two cases may occur. If I is bounded then it is of the form (m,m') or $(m,m']$ in this case we define[11] $\lambda(q,y) = (\frac{1}{2}(m+m'),c)$. Finally if I has no minimum and is unbounded it is of the form (m,∞) and in this case we

[11] Let us recall that every o-minimal ordered group is torsion free and divisible (see [PS86]), this implies there exists a unique y satisfying $y + y = (m+m')$, which we note $\frac{1}{2}(m+m')$.

define $\lambda(q,y) = (m+1, c)$. We summarize[12] the definition of λ on S_l as follows:

$$\lambda(q,y) = \begin{cases} \big(\min(I), c\big) & \text{if } \varphi_1(y) \\ \big(\frac{1}{2}\big(\inf(I) + \sup(I)\big), c\big) & \text{if } \varphi_2(y) \\ \big(\inf(I)+1, c\big) & \text{otherwise} \end{cases}$$

where $\varphi_1(y)$ is a formula which is true if and only if I (or $\mathsf{Time}(y)$) has a minimum and $\varphi_2(y)$ is a formula which is true if and only if I has no minimum and is bounded. Thus clearly λ is definable.

Since there are finitely many $P \in \mathsf{Sup}(\mathcal{P}_\mathcal{A})$, we can conclude that λ is definable. □

Remark 6.9. Note that the memoryless strategy given by Theorem 6.8 is computable if $\pi^*_{\mathsf{partial}}(\mathsf{Goal})$ is.

Remark 6.10. Let us notice that in the case of timed automata dynamics (described in Example 3.2), our definable strategies correspond to the realizable strategies computed in [BCFL04].

6.6. **Decidability Result.** Theorem 6.8 is an existential result. It claims that given an o-minimal game, there exists a definable memoryless strategy for each $y \in \pi^*_{\mathsf{partial}}(\mathsf{Goal})$, and by Theorem 6.5 we know that $\mathsf{Sup}(\mathcal{P}_\mathcal{A})$ is finite. The conclusion of the previous subsection is that given an o-minimal game there exists a definable memoryless winning strategy for each $y \in \pi^*_{\mathsf{partial}}(\mathsf{Goal})$.

In general, Theorem 6.8 does not allow to conclude that the control problem in an $\mathcal{M}$-structure is decidable. Indeed it depends on the decidability of $\mathsf{Th}(\mathcal{M})$. We can state the following theorem:

Theorem 6.11. *Let $\mathcal{M}$ be an o-minimal structure such that $\mathsf{Th}(\mathcal{M})$ is decidable and $\mathcal{C}$ a class of $\mathcal{M}$-automata. Then the control problem under partial (resp. perfect) observation in class $\mathcal{C}$ is decidable. Moreover if $\mathcal{A} \in \mathcal{C}$, the set of winning states $\pi^*_{\mathsf{partial}}(\mathsf{Goal})$ (resp. $\pi^*_{\mathsf{perfect}}(\mathsf{Goal})$) under partial (resp. perfect) observation is computable and a memoryless winning strategy can be effectively computed for each $(q,y) \in \pi^*_{\mathsf{partial}}(\mathsf{Goal})$ (resp. $\pi^*_{\mathsf{perfect}}(\mathsf{Goal})$).*

Proof. By Proposition 6.7, for each $\mathcal{A} \in \mathcal{C}$, $\mathsf{Sup}(\mathcal{P}_\mathcal{A})$ is a definable finite partition respecting Goal. Moreover by Proposition 6.5, $\mathsf{Sup}(\mathcal{P}_\mathcal{A})$ is stable under π_{partial}. Hypothesis of Corollary 5.6 are thus satisfied and we get that the control problem in class $\mathcal{C}$ is decidable and that the winning states of a game $\mathcal{A} \in \mathcal{C}$ are computable. Moreover Theorem 6.8 ensures that a memoryless strategy can be effectively defined from such winning states. □

[12] Let us notice that the way we extract a single point from $\mathsf{Time}(y)$ is nothing more than the *curve selection* for o-minimal expansions of ordered abelian groups, see [vdD98, chap.6].

Remark 6.12. $\langle \mathbb{R}, <, +, 0, 1\rangle$ and $\langle \mathbb{R}, <, +, \cdot, 0, 1\rangle$ are examples of o-minimal structures with decidable theory and so o-minimal games based on theses structures can be solved by Theorem 6.11.

Remark 6.13. In this paper we did not distinguish Zeno behaviours. In particular, in our framework, if the environment has a strategy that prevents the game to reach the Goal locations by blocking time, we say that the controller loses the game. In the framework of timed automata, an *ad-hoc* solution to this *problem of Zenoness* has been proposed in [AFH+03]. However, due to the strong reset conditions of o-minimal hybrid systems, the method of [AFH+03] cannot be easily applied to our framework, but this problem is somehow orthogonal to ours.

7. Conclusion

In this paper we have studied games based on dynamical systems with general dynamics, both under a prefect and a partial observation of the dynamics. Under the first hypothesis, we have shown that time-abstract bisimulation is not fine enough to solve these games, which is a major difference with the case of timed automata. By means of an encoding of trajectories by words, we have obtained a good abstraction for control problems (with reachability winning conditions, but it applies also to basic safety winning conditions). We have finally provided decidability and computability results for o-minimal games under both perfect and partial observation hypothesis. Our technique applies to timed automata, and we recover decidability of timed games [AMPS98], as well as the construction of winning strategies [BCFL04] as side results.

Acknowledgment

The two first authors have been partly supported by the ESF project GASICS. The first author has been partly supported by the project DOTS (ANR-06-SETI-003) and by the EU project QUASIMODO. The second author has been partly supported by a grant from the National Bank of Belgium and by a FRFC grant: 2.4530.02.

References

[Acz88] Peter Aczel. *Non-Well-Founded Sets*, volume 14 of *CSLI Lecture Notes*. Center for the Study of Language and Information, Stanford University, 1988.

[AD90] Rajeev Alur and David Dill. Automata for modeling real-time systems. In *Proc. 17th International Colloquium on Automata, Languages and Programming (ICALP'90)*, volume 443 of *Lecture Notes in Computer Science*, pages 322–335. Springer, 1990.

[AD94] Rajeev Alur and David Dill. A theory of timed automata. *Theoretical Computer Science*, 126(2):183–235, 1994.

[AFH+03] Luca de Alfaro, Marco Faella, Thomas A. Henzinger, Rapuk Majumdar, and Mariëlla Stoelinga. The element of surprise in timed games. In *Proc. 14th International Conference on Concurrency Theory (CONCUR'03)*, volume 2761 of *Lecture Notes in Computer Science*, pages 142–156. Springer, 2003.

[AHLP00] Rajeev Alur, Thomas A. Henzinger, Gerardo Lafferriere, and George J. Pappa. Discrete abstractions of hybrid systems. *Proc. of the IEEE*, 88:971–984, 2000.

[ALM05] Rajeev Alur, Salvatore La Torre, and P. Madhusudan. Perturbed timed automata. In *Proc. 8th International Workshop on Hybrid Systems: Computation and Control (HSCC'05)*, volume 3414 of *Lecture Notes in Computer Science*, pages 70–85. Springer, 2005.

[AMPS98] Eugene Asarin, Oded Maler, Amir Pnueli, and Joseph Sifakis. Controller synthesis for timed automata. In *Proc. IFAC Symposium on System Structure and Control*, pages 469–474. Elsevier Science, 1998.

[ASY01] Eugene Asarin, Gerardo Schneider, and Sergio Yovine. On the decidability of the reachability problem for planar differential inclusions. In *Proc. 4th International Workshop on Hybrid Systems: Computation and Control (HSCC'01)*, volume 2034 of *Lecture Notes in Computer Science*, pages 89–104. Springer, 2001.

[AVW03] André Arnold, Aymeric Vincent, and Igor Walukiewicz. Games for synthesis of controllers with partial observation. *Theoretical Computer Science*, 1(303):7–34, 2003.

[BBC06] Patricia Bouyer, Thomas Brihaye, and Fabrice Chevalier. Control in o-minimal hybrid systems. In *Proc. 21st Annual IEEE Symposium on Logic in Computer Science (LICS'06)*, pages 367–378. IEEE Computer Society Press, 2006.

[BCFL04] Patricia Bouyer, Franck Cassez, Emmanuel Fleury, and Kim G. Larsen. Optimal strategies in priced timed game automata. In *Proc. 24th Conference on Foundations of Software Technology and Theoretical Computer Science (FST&TCS'04)*, volume 3328 of *Lecture Notes in Computer Science*, pages 148–160. Springer, 2004.

[BDMP03] Patricia Bouyer, Deepak D'Souza, P. Madhusudan, and Antoine Petit. Timed control with partial observability. In *Proc. 15th International Conference on Computer Aided Verification (CAV'03)*, volume 2725 of *Lecture Notes in Computer Science*, pages 180–192. Springer, 2003.

[BM05] Thomas Brihaye and Christian Michaux. On the expressiveness and decidability of o-minimal hybrid systems. *Journal of Complexity*, 21(4):447–478, 2005.

[BMRT04] Thomas Brihaye, Christian Michaux, Cédric Rivière, and Christophe Troestler. On o-minimal hybrid systems. In *Proc. 7th International Workshop on Hybrid Systems: Computation and Control (HSCC'04)*, volume 2993 of *Lecture Notes in Computer Science*, pages 219–233. Springer, 2004.

[Bri06] Thomas Brihaye. *Verification and Control of O-Minimal Hybrid Systems and Weighted Timed Automata*. PhD thesis, Université de Mons-Hainaut, Belgium, 2006.

[Bri07] Thomas Brihaye. Words and bisimulations of dynamical systems. *Discrete Math. Theor. Comput. Sci.*, 9(2):11–31, 2007.

[Cau95] Didier Caucal. *Bisimulation of Context-Free Grammars and of Pushdown Automata*, volume 53 of *CSLI Lecture Notes*, pages 85–106. Stanford University, 1995.

[dAHM01] Luca de Alfaro, Thomas A. Henzinger, and Rupak Majumdar. Symbolic algorithms for infinite-state games. In *Proc. 12th International Conference on Concurrency Theory (CONCUR'01)*, volume 2154 of *Lecture Notes in Computer Science*, pages 536–550. Springer, 2001.

[Dav99] Jennifer M. Davoren. Topologies, continuity and bisimulations. *Informatique Théorique et Applications*, 33(4-5):357–382, 1999.

[GTW02] Erich Grädel, Wolfgang Thomas, and Thomas Wilke, editors. *Automata, Logics, and Infinite Games: A Guide to Current Research*, volume 2500 of *Lecture Notes in Computer Science*. Springer, 2002.

[Hen95] Thomas A. Henzinger. Hybrid automata with finite bisimulations. In *Proc. 22nd International Colloquium on Automata, Languages and Programming (ICALP'95)*, volume 944 of *Lecture Notes in Computer Science*, pages 324–335. Springer, 1995.

[Hen96] Thomas A. Henzinger. The theory of hybrid automata. In *Proc. 11th Annual Symposim on Logic in Computer Science (LICS'96)*, pages 278–292. IEEE Computer Society Press, 1996.

[HHM99] Thomas A. Henzinger, Benjamin Horowitz, and Rupak Majumdar. Rectangular hybrid games. In *Proc. 10th International Conference on Concurrency Theory (CONCUR'99)*, volume 1664 of *Lecture Notes in Computer Science*, pages 320–335. Springer, 1999.

[HKPV98] Thomas A. Henzinger, Peter W. Kopke, Anuj Puri, and Pravin Varaiya. What's decidable about hybrid automata? *Journal of Computer and System Sciences*, 57(1):94–124, 1998.

[Hod97] Wilfrid Hodges. *A Shorter Model Theory*. Cambridge University Press, 1997.

[KPS86] Julia F. Knight, Anand Pillay, and Charles Steinhorn. Definable sets in ordered structures ii. *Transactions of the American Mathematical Society*, 295(2):593–605, 1986.

[KRS07] Felix Klaedtke, Stefan Ratschan, and Zhikun She. Language-based abstraction refinement for hybrid system verification. In *Proc. 8th International Conference on Verification, Model Checking,* *and Abstract Interpretation*, volume 4349 of *Lecture Notes in Computer Science*, pages 151–155. Springer-Verlag, 2007.

[KV04] Margarita V. Korovina and Nicolai Vorobjov. Pfaffian hybrid systems. In *Proc. 18th International Workshop on Computer Science Logic (CSL'04)*, volume 3210 of *Lecture Notes in Computer Science*, pages 430–441. Springer, 2004.

[KV06] Margarita V. Korovina and Nicolai Vorobjov. Upper and lower bounds on sizes of finite bisimulations of Pfaffian hybrid systems. In *CiE*, volume 3988 of *Lecture Notes in Computer Science*, pages 267–276. Springer, 2006.

[LPS00] Gerardo Lafferriere, George J. Pappas, and Shankar Sastry. O-minimal hybrid systems. *Mathematics of Control, Signals, and Systems*, 13(1):1–21, 2000.

[Mil89] Robert Milner. *Communication and Concurrency.* Prentice Hall International, 1989.

[PS86] Anand Pillay and Charles Steinhorn. Definable sets in ordered structures. *Transactions of the American Mathematical Society*, 295(2):565–592, 1986.

[Pur98] Anuj Puri. Dynamical properties of timed automata. In *Proc. 5th International Symposium on Formal techniques in Real-Time and Fault-Tolerant Systems (FTRTFT'98)*, volume 1486 of *Lecture Notes in Computer Science*, pages 210–227. Springer, 1998.

[vdD96] Lou van den Dries. O-minimal structures. In *Proc. Logic, From Foundations to Applications*, Oxford Science Publications, pages 137–185. Oxford University Press, 1996.

[vdD98] Lou van den Dries. *Tame Topology and O-Minimal Structures*, volume 248 of *London Mathematical Society Lecture Note Series.* Cambridge University Press, 1998.

[Wil96] Alex J. Wilkie. Model completeness results for expansions of the ordered field of real numbers by restricted Pfaffian functions and the exponential function. *Journal of the American Mathematical Society*, 9(4):1051–1094, 1996.

[WT97] Howard Wong-Toi. The synthesis of controllers for linear hybrid automata. In *Proc. 36th IEEE Conference on Decision and Control*, pages 4607–4612. IEEE Computer Society Press, 1997.