Centre Fédéré en Vérification

Technical Report number 2006.62

Control in o-minimal hybrid systems

Patricia Bouyer, Thomas Brihaye, Fabrice Chevalier

This work was partially supported by a FRFC grant: 2.4530.02

http://www.ulb.ac.be/di/ssd/cfv

Control in o-minimal hybrid systems

Patricia Bouyer[1]*, Thomas Brihaye[2]†, Fabrice Chevalier[1]*

[1] LSV - CNRS & ENS de Cachan
61, avenue du Président Wilson, 94230 Cachan, France
e-mails: {bouyer,chevalie}@lsv.ens-cachan.fr
[2] Université de Mons-Hainaut, Institut de Mathématique
6, avenue du Champ de Mars, 7000 Mons, Belgium
e-mail: thomas.brihaye@umh.ac.be

Abstract. In this paper, we consider the control of general hybrid systems. We show that, surprisingly, time-abstract bisimulation is not fine enough for solving such a problem. Conversely, we show that suffix equivalence is a correct abstraction for that problem. We apply this equivalence to o-minimal hybrid systems and get decidability and computability results in this framework.

1 Introduction

Control of 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 [1], have been intensively studied [13, 14], and decidable subclasses of hybrid systems have been drawn like initialized rectangular hybrid automata [14] or o-minimal hybrid automata [18]. More recently, the control of hybrid systems has appeared as a new interesting and active field of research, and many results have already been obtained, like the (un)decidability of control problems for hybrid automata [15], or (semi-)algorithms for solving such problems [11]. 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 [18] as an interesting class of systems (see [23] 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

* Work partly supported by ACI Cortos, a program of the French ministry of research.
† This author is supported by the following research programs: FRFC 2.4.530.02., FRFC 2.4.564.02, Modnet MRTN-CT-2004-512234 and by a grant from the National Bank of Belgium.

allows to decouple the continuous and discrete components of the hybrid system (see [18]). Thus, properties of a global o-minimal system can be deduced directly from properties of the continuous behaviors of the system. Since the introductory paper [18], several works have considered o-minimal hybrid systems [10, 8, 7, 17], mostly focusing on abstractions of such systems, on reachability properties, and on bisimulation properties.

Word encoding. In [8], 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 [7]). Let us mention that this technique has been used in [17] in order to provide an exponential bound on the size of the finite bisimulation in the case of pfaffian hybrid systems. Different word encoding techniques have been studied in a wider context in [6]. 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 [6] for computing a time-abstract bisimulation, and it terminates in the case of o-minimal hybrid systems (under some word uniqueness hypothesis[1]).

Contributions of this paper. In this paper, we focus on the control of hybrid systems, and 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 [2]. Game bisimulation is correct for discrete infinite-state games [11]. On the contrary, we show that the time-abstract bisimulation is not correct for solving control problems (with a reachability objective): we exhibit a system in which two states are time-abstract bisimilar, but one of the states is winning and the other is not winning. Using the word encoding of trajectories of [6], we prove that two states having the same suffixes in this encoding are equivalently winning or losing (this is a stronger condition than for the time-abstract bisimulation). We finally focus on o-minimal hybrid games and prove that, under the assumption that the theory of the underlying o-minimal structure is decidable and assuming that each state has a unique suffix, the control problem can be solved and that winning states and winning strategies can be computed. Note that this unique suffix assumption is not that restrictive as it encompasses the assumptions of [18] where continuous dynamics are time-deterministic.

Plan of the paper. Section 2, we define the hybrid games we will consider, and we show that time-abstract bisimulation is not correct for solving them. The word encoding technique is presented in Section 3 and used in Section 4 to present a general framework for solving hybrid games. We apply these results in Section 5 for computing winning states and winning strategies in o-minimal hybrid games.

All proofs can be found in the appendix.

[1] Notice that when this word uniqueness assumption is relaxed, the reachability problem becomes undecidable for o-minimal hybrid systems (see [5]).

2 Games over dynamical systems

2.1 Dynamical systems

Let $\mathcal{M}$ be a structure. In this paper when we say that some relation, subset or function is *definable*, we mean it is first-order definable in the sense of the structure $\mathcal{M}$. A general reference for first-order logic is [16]. 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.*** $\mathcal{M} = \langle M, +, 0, 1, <, ...\rangle$ and without loss of generality we assume that $0 < 1$.

Definition 2.1. A *dynamical system* is a pair $(\mathcal{M}, \gamma)$ where:

- $\mathcal{M} = \langle M, +, 0, 1, <, ...\rangle$ is an expansion of an ordered group,
- $\gamma : V_1 \times M^+ \to V_2$ is a function
  (where $M^+ = \{m \in M \mid m \geqslant 0\}$, $V_1 \subseteq M^{k_1}$, and $V_2 \subseteq M^{k_2}$).[2]

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

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

The definition of *dynamical systems* encompasses a lot of different behaviors. Let us give a simple example.

Example 2.2. We can recover the continuous dynamics of *timed automata* (see [1]). 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 $\gamma(x_1, ..., x_n, t) = (x_1 + t, ..., x_n + t)$.

Definition 2.3. 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 [18] for example).

Definition 2.4. Given $(\mathcal{M}, \gamma)$ a dynamical system, we define a *transition system* $T_\gamma = (Q, \Sigma, \rightarrow_\gamma)$ *associated with the dynamical system* by:

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

$$\exists x \in V_1,\ \exists t_1, t_2 \in M^+,\ \big(t_1 \leqslant t_2,\ \gamma(x, t_1) = y_1,\ \gamma(x, t_2) = y_2 \text{ and } t = t_2 - t_1\big)$$

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

2.2 $\mathcal{M}$-games

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 real-time 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 an $\mathcal{M}$-structure*.

Definition 2.5 ($\mathcal{M}$-automaton). An $\mathcal{M}$-*automaton* $\mathcal{A}$ is a tuple $(\mathcal{M}, Q, \mathsf{Goal}, \Sigma, \delta, \gamma)$ where $\mathcal{M} = (M, +, <, \cdots)$ 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 dynamic $\gamma_q : V_1 \times M^+ \to V_2$ definable in $\mathcal{M}$.

We use a general definition for resets: a reset R is indeed a general function from V_2 to 2^{V_2}, which may for example 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:

- the set S of states is $Q \times V_2$;
- the set Γ of labels is $M^+ \cup \Sigma$;
- 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 $y' \in R(y)$,
  - $e \in M^+$, $q = q'$, 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, like classically in timed automata. We will also need more precise notions of transitions. When $(q, y) \xrightarrow{t'} (q, y')$ with $t' \in M^+$, this is due to some choice of $(x, t) \in V_1 \times M^+$ such that $\gamma_q(x, t) = y$. We say that $(q, y) \xrightarrow{t'}_{x,t} (q, y')$ if $(q, y) \xrightarrow{t'} (q, y')$, $\gamma_q(x, t) = y$ and $\gamma_q(x, t + t') = y'$. To ease the reading of the paper, we will sometimes write $(q, x, t, y) \xrightarrow{t'} (q, x, t + t', y')$ for $(q, y) \xrightarrow{t'}_{x,t} (q, y')$. We say that an action $(d, a) \in M^+ \times \Sigma$ is enabled in a state (q, x, t, y) if there exists a (q', x', t', y') such that $(q, x, t, y) \xrightarrow{d}\xrightarrow{a} (q', x', t', y')$. We then write $(q, x, t, y) \xrightarrow{d,a} (q', x', t', y')$.

A *run* of $\mathcal{A}$ is a finite or infinite sequence $(q_0, x_0, t'_0, y_0) \xrightarrow{t_1,a_1} (q_1, x_1, t'_1, y_1) \ldots$ where for every i, $(q_i, y_i) \xrightarrow{t_i}_{t'_i,x_i} (q_i, y'_i) \xrightarrow{a} (q_{i+1}, y_{i+1})$. Such a run is said *winning* if $q_i \in \mathsf{Goal}$ for some i. We note $\mathsf{Runs}(\mathcal{A}, (q, x, t, y))$ (resp. $\mathsf{Runs}_f(\mathcal{A}, (q, x, t, y))$) the set of (finite) runs starting in (q, x, t, y), and we note $\mathsf{Runs}(\mathcal{A})$ the set of all runs in $\mathcal{A}$. If ρ is a finite run $(q_0, x_0, t'_0, y_0) \xrightarrow{t_1,a_1} \ldots \xrightarrow{t_n,a_n} (q_n, x_n, t'_n, y_n)$ we define $last(\rho) = (q_n, x_n, t'_n, y_n)$.

Definition 2.6 ($\mathcal{M}$-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.

Without loss of generality, we suppose that there is a loop labeled by a controllable action on every state of Goal.

Definition 2.7 (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 $\mathsf{Runs}(\mathcal{A})$, $\lambda(\rho)$ is enabled in $last(\rho)$.

The strategy tells what needs to be done for controlling the system: at each instant it tells what delay we need to wait and which action needs to be done after this delay. Note then 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 deterministic).

Let $\rho = (q_0, x_0, t'_0, y_0) \xrightarrow{t_1, a_1} \ldots$ be an 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 $\lambda(\rho_i) = (t, a)$ then either $t_{i+1} = t$ and $a_{i+1} = a$, or $t_{i+1} \leqslant t$ and $a_{i+1} \in \Sigma_u$. A run ρ is said *maximal* if it is infinite or if it is finite ending in (q, x, t, y) and satisfies that for all $t' \geqslant 0$, for all $a \in \Sigma$, "$(q, x, t, y) \xrightarrow{t', a}$" implies $a \in \Sigma_u$. A strategy λ is *winning from a state (q,x,t,y)* if all maximal runs starting in (q, x, t, y) compatible with λ are winning.

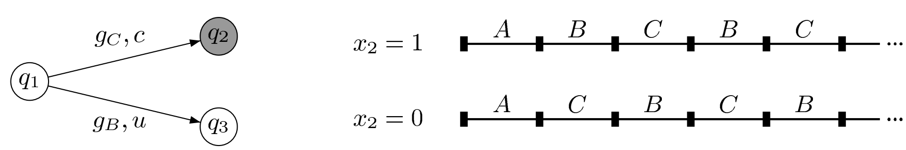
We can now define the control problem we will study.

Problem 2.8 (Control problem 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 if there exists a winning strategy in $\mathcal{A}$ from (q, y).*

2.3 $\mathcal{M}$-game and bisimulation

Time-abstraction bisimulation [10, 2, 13] is a sufficient behavioral relation to check reachability properties of timed systems, and in particular of $\mathcal{M}$-automata [6]. When considering control problems, we will see that this is no more the case in general.

Example 2.9. Let us consider the $\mathcal{M}$-game $\mathcal{A} = (\mathcal{M}, Q, \mathsf{Goal}, \Sigma, \delta, \gamma)$ where $\mathcal{M} = \langle \mathbb{R}, <, +, 0, 1, \equiv_1 \rangle$ ($\equiv_1$ denotes the "modulo 1" relation), $Q = \{q_1, q_2, q_3\}$, $\mathsf{Goal} = \{q_2\}$, $\Sigma = \{c, u\}$. The dynamic in q_1, $\gamma_{q_1} : \mathbb{R}^+ \times \{0,1\} \times \mathbb{R}^+ \to \mathbb{R}^+ \times \{0,1\}$ 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

Fig. 1. Time-abstract bisimulation does not preserve winning states

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

We consider the partition depicted on Figure 1(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.

Proposition 2.10. *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 $\mathcal{A}$.*

3 Suffix and dynamical type

In this section we explain how to encode trajectories of dynamical systems through words. This technique was introduced in [8, 7] in order to study o-minimal hybrid systems. We focus on the *suffix partition* introduced in [6].

First let us define the notion of *word* in this general (possibly uncountable) context. This definition is inspired from [21, 9, 20].

Definition 3.1. Given $\mathcal{P}$ a finite set (called the *alphabet*), M a totally ordered set, a *word ω on $\mathcal{P}$* is a function from M to $\mathcal{P}$; the word ω is also denoted in a sequence-like notation by $(\omega_i)_{i \in M}$ where $\omega_i \in \mathcal{P}$ is the image of the element i under the function ω.

Given $\omega : M \to \mathcal{P}$ a word on $\mathcal{P}$, a *suffix of ω* is a sub-word $\omega_s : M' \to \mathcal{P}$ of ω such that $M' = \{t \in M^+ \mid t \geqslant t_0\}$ or $M' = \{t \in M^+ \mid t > t_0\}$ for some $t_0 \in M$.

We are now ready to build words associated with trajectories. Given $(\mathcal{M}, \gamma)$ a dynamical system and $\mathcal{P}$ a finite partition of V_2, given $x \in V_1$ we associate a word with the trajectory Γ_x in the following way. We consider the sets $\{t \in M^+ \mid \gamma(x, t) \in P\}$ for $P \in \mathcal{P}$. This gives a partition of the time M^+. In order to define a word on $\mathcal{P}$ associated with the trajectory determined by x, we need to define the set of intervals $\mathcal{F}_x = \big\{I \;\big|\; I \text{ is a time interval or a point and is maximal for the property } \exists P \in \mathcal{P},\ \forall t \in I,\ \gamma(x, t) \in P\big\}$. 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 Γ_x* denoted ω_x.

Definition 3.2. 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 $V_1 \times M^+$, 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

$$\mathcal{F}_{(x,t)} = \big\{ I \cap \{t' \in M^+ \mid t' \geqslant t\} \bigm| I \in \mathcal{F}_x \big\}.$$

Let us notice that given (x,t) a point of the space-time $V_1 \times M^+$ 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 dynamical type* of a point y w.r.t. $\mathcal{P}$.

Definition 3.3. Given a dynamical system $(\mathcal{M},\gamma)$, a finite partition $\mathcal{P}$ of V_2, a point $y \in V_2$ the *suffix dynamical type 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 $\mathsf{Suf}_{\mathcal{P}}(y_1) = \mathsf{Suf}_{\mathcal{P}}(y_2)$, and we note $y_1 \equiv_{\mathcal{P}} y_2$. We denote by $\mathsf{Suf}\,(\mathcal{P})$ the partition induced by the suffix equivalence $(\equiv_{\mathcal{P}})$.

We say that a partition $\mathcal{P}$ is *suffix-stable* if $\mathsf{Suf}(\mathcal{P}) = \mathcal{P}$.

To understand the word encoding technique, let us illustrate it on an example.

Example 3.4. We consider the dynamical system $(\mathcal{M},\gamma)$ where $\mathcal{M} = \langle \mathbb{R}, +, \cdot, 0, 1, <, \sin_{|[0,2\pi]}, \cos_{|[0,2\pi]} \rangle$ [4] and $\gamma : \mathbb{R}^2 \times [0,2\pi] \times \mathbb{R} \to \mathbb{R}^2$ is defined as follows.

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

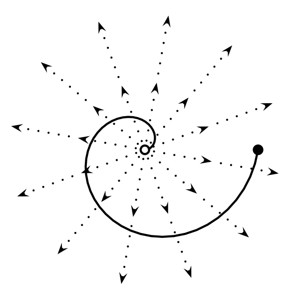


Fig. 2. The dynamical system of the spiral

We associate with this dynamical system the partition $\mathcal{P} = \{A, B, C\}$ where $A = \{(0,0)\}$, $B = \{\big(\theta\cos(\theta), \theta\sin(\theta)\big) \mid 0 < \theta \leqslant 2\pi\}$ and $C = (\mathbb{R})^2 \setminus (A \cup C)$. Let us call

[4] $\sin_{|[0,2\pi]}$ and $\cos_{|[0,2\pi]}$ correspond to the sinus and cosinus functions restricted to the segments $[0,2\pi]$.

piece B *the spiral*. There are four dynamical types for this system: $\{ACBC\}$, $\{CBC\}$, $\{BC\}$ and $\{C\}$. Let us notice that though the dynamical system is infi nitely branching in $(0, 0)$, there is a unique suffi x associated with each point y of the output space.

4 Solving an $\mathcal{M}$-game

In this section we present a procedure to compute the set of winning states for an $\mathcal{M}$-game. We then show that if a partition is *suffix-stable*, the procedure can be performed symbolically on pieces of the partition.[5] The procedure described is not always effective and we will point out specifi c $\mathcal{M}$-structures for which each step of the procedure is computable.

4.1 Controllable predecessors

As for classical reachability games [12], 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 $A \subseteq Q \times V_2$ and $a \in \Sigma$ we defi ne the controllable and uncontrollable discrete predecessors as follows:

$$\mathsf{cPred}(A) = \left\{ (q, y) \in Q \times V_2 \;\middle|\; \begin{array}{l} \exists c \in \Sigma_c,\ \text{c is enabled in } (q, y), \\ \text{and } \forall (q', y') \in Q \times V_2, \\ (q, y) \xrightarrow{c} (q', y') \Rightarrow (q', y') \in A \end{array} \right\}$$

$$\mathsf{uPred}(A) = \left\{ (q, y) \in Q \times V_2 \;\middle|\; \begin{array}{l} \exists u \in \Sigma_u,\ \exists (q', y') \in Q \times V_2 \\ \text{s.t. } (q, y) \xrightarrow{u} (q', y') \text{ and } (q', y') \in A \end{array} \right\}$$

As for timed and hybrid games [3, 15], we also defi ne a *safe* time predecessor of a set A w.r.t. a set B: a state (q, y) is in $\mathsf{Pred}_t(A, B)$ if, by letting time elapse, one reaches $(q', y') \in A$, avoiding B. Formally the operator Pred_t is defi ned as follows:

$$\mathsf{Pred}_t(A, B) = \left\{ (q, y) \in Q \times V_2 \;\middle|\; \begin{array}{l} \forall (x, t) \in V_1 \times M^+,\ \exists t' \in M^+ \text{ s.t.} \\ (q, y) \xrightarrow{t'}_{x,t} (q', y'),\ (q', y') \in A, \\ \text{and } \mathsf{Post}^{q,x}_{[t,t+t']} \subseteq \overline{B} \end{array} \right\}$$

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

The *controllable predecessor* operator is then defi ned as:

$$\pi(A) = A \cup \mathsf{Pred}_t(\mathsf{cPred}(A), \mathsf{uPred}(\overline{A}))$$

Intuitively, a state (q, y) is in $\pi(A)$ whenever either it is already in A or there is a way of waiting some amount of time, and of performing a controllable action to enter A, and no uncontrollable action leads outside A.

[5] The effectivity of the computation will be discussed later.

Remark 4.1. Note that the operator π is definable in any expansion of an ordered group. Hence, if A is definable, so is $\pi(A)$.

We will compute the set of winning states by iterating the operator π: denoting $\pi^*(\mathsf{Goal}) = \bigcup_{k \geqslant 0} \pi^k(\mathsf{Goal})$, we will show that the set of winning states for the game is precisely $\pi^*(\mathsf{Goal})$. This will help getting further effective definability and computability results of winning states and winning strategies under some assumption on the underlying structure.

Proposition 4.2. *Let $\mathcal{A} = (\mathcal{M}, Q, \mathsf{Goal}, \Sigma, \delta, \gamma)$ be an $\mathcal{M}$-game, and $(q, y) \in Q \times V_2$. Then, $(q, y) \in \pi^*(\mathsf{Goal})$ iff there is a winning strategy in $\mathcal{A}$ from (q, y).*

We now deduce an algorithmic result from proposition 4.2. The set of winning states is $\pi^*(\mathsf{Goal})$ but this does not imply that we can compute this set as many $\mathcal{M}$-structure structures are already intrinsically undecidable. The following corollary states that if some conditions on the structure and π are satisfied, then this procedure provides an algorithmic solution to the control problem:

Corollary 4.3. *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[6]*, and stable by π. Then the control problem in the class $\mathcal{C}$ is decidable. Moreover if $\mathcal{A} \in \mathcal{C}$, the set of winning states of $\mathcal{A}$ is computable.*

4.2 Stability of Suf $(\mathcal{P})$

In section 2.3, we have presented a counter-example which showed that bisimulation was not correct to solve control problems; the main reason was that the partition induced by bisimilarity was not stable under the operator π.

We now present a sufficient condition for a partition to be stable under the operator π: we require that the partition is stable under cPred and uPred to handle the discrete part of the automaton and we show that the stability by suffix is fine enough to ensure a good continuous behavior w.r.t. control problems.

Proposition 4.4. *Let $\mathcal{A}$ be an $\mathcal{M}$-game, $\mathcal{P}$ be a partition of $Q \times V_2$ and π be the controllable predecessor operator. If $\mathcal{P}$ respects* Goal*, is stable under* cPred*,* uPred *and suffix-stable, then $\mathcal{P}$ is stable under the operator π.*

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 $y_1 \in \pi(X)$, for some $X \in \mathcal{P}$ then $y_2 \in \pi(X)$.

Since $y_1 \in \pi(X)$, for all $(x, t) \in V_1 \times M^+$ such that $\gamma_q(x, t) = y_1$, there exists y_1' such that $y_1 \xrightarrow{t_1'}_{x,t} y_1'$ for some $t_1' \in M^+$, $y_1' \in \mathsf{cPred}(X)$, and $\mathsf{Post}^{q,x}_{[t,t+t_1']}(X) \subseteq \overline{\mathsf{uPred}(\overline{X})}$.

Let $\omega_{(x,t)}(y_1)$ be a suffix of y_1 associated to x and t. In term of words the three previous conditions mean that there exists a prefix of $\omega_{(x,t)}(y_1)$ whose last letter is in

[6] *i.e.* Goal is a union of pieces of $\mathcal{P}$

$\mathsf{cPred}(X)$ and with no occurrence of letters of $\mathsf{uPred}(\overline{X})$ (this has a meaning as by hypothesis $\mathsf{cPred}(X)$ and $\mathsf{uPred}(\overline{X})$ are unions of pieces of $\mathcal{P}$). Let us call $\omega^p_{(x,t)}(y_1)$ this prefix.

Since $\mathcal{P} = \mathsf{Suf}(\mathcal{P})$ and y_1 and y_2 belong to the same piece of $\mathcal{P}$, we have that $\mathsf{Suf}_{\mathcal{P}}(y_1) = \mathsf{Suf}_{\mathcal{P}}(y_2)$. Let $(x,t) \in V_1 \times M^+$ such that $\gamma(x,t) = y_2$ and let $\omega_{(x,t)}(y_2)$ be the suffix of y_2 associated to x and t. As $\mathsf{Suf}_{\mathcal{P}}(y_1) = \mathsf{Suf}_{\mathcal{P}}(y_2)$, $\omega_{(x,t)}(y_2) = \omega_{(x',t')}(y_1)$ for some (x',t') such that $\gamma_q(x',t') = y_1$, so the prefix $\omega^p_{(x',t')}(y_1)$ is a prefix of $\omega_{(x,t)}(y_2)$.

So we can find $y_2' \in \mathsf{cPred}(X)$ such that $y_2 \xrightarrow{t_2'}_{x,t} y_2'$ for some $t_2' \in M^+$, $y_2' \in \mathsf{cPred}(X)$, and $\mathsf{Post}^{q,x}_{[t,t+t_2']}(X) \subseteq \overline{\mathsf{uPred}(\overline{X})}$. Thus $y_2 \in \pi(X)$. □

Remark 4.5. The results of this section permit to recover the results of [3] about control of timed automata. We consider the classical finite partition of timed automata that induces the region graph (see [1]). Let us call $\mathcal{P}_R$ this partition, and notice that $\mathcal{P}_R$ is definable in $\langle \mathbb{R}, <, +, 0, 1\rangle$. The equivalence relation induced by $\mathcal{P}_R$ is a time-abstract bisimulation. Hence in particular $\mathcal{P}_R$ is stable under the action of cPred and uPred. By Example 2.2 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. One can easily be convinced that $\mathsf{Suf}(\mathcal{P}_R) = \mathcal{P}_R$. We thus conclude that $\mathcal{P}_R$ is stable under the action of π (see Proposition 4.4). Hypotheses of corollary 4.3 are thus satisfied and we get the computability of winning states in timed games [3] as a side result by computing $\pi^*(\mathsf{Goal})$.

5 Case of 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) [18].

We first briefly recall definitions and results related to o-minimality. The reader interested in o-minimality should refer to [23] for further results and an extensive bibliography on this subject. Then we focus on o-minimal structures with a decidable theory in order to obtain decidability and computability results.

Definition 5.1. An extension of an ordered structure $\mathcal{M} = \langle M, <, ...\rangle$ 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 $\langle M, <\rangle$. The following are examples of o-minimal structures.

Example 5.2. There are many 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.

5.1 Generalities on o-minimal games

Definition 5.3. 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[7] $Q \times 2^{V_2} \times \Sigma \times 2^{V_2} \times Q$.

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 partition of the state space $S = Q \times V_2$ which respects all guards and resets in $\mathcal{A}$. Note that $\mathcal{P}_\mathcal{A}$ is a finite definable partition of S.

Clearly the partition $\mathcal{P}_\mathcal{A}$ respects the guards, the resets and Goal. Moreover due to the strong reset condition we have that $\mathcal{P}_\mathcal{A}$ is stable under the action of cPred and uPred. This holds by the same argument that allows to decouple the continuous and discrete components of the hybrid system in [18]. 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 cPred and uPred.

O-minimal games are *o-minimal hybrid systems* (as defined in [7]). With slight adaptations of Lemma 4.13 and Theorem 4.18 of [7], we can easily deduce the following result.

Theorem 5.4 ([7]). *Let $\mathcal{A}$ be an o-minimal game. If there exists a unique suffix on $\mathcal{P}_\mathcal{A}$ associated with each $(q, y) \in Q \times V_2$ then the partition $\mathsf{Suf}(\mathcal{P}_\mathcal{A})$ is finite and definable. Moreover $\mathsf{Suf}(\mathcal{P}_\mathcal{A})$ is a time-abstract bisimulation.*

In particular, every piece $A \in \mathsf{Suf}(\mathcal{P})$ is definable in the structure $\mathcal{M}$.

Remark 5.5. Note also that this "unique suffix" assumption is reasonable as there already exist infinite time-abstract bisimulations when two suffixes are allowed (see the *torus example* in [8]), and reachability in o-minimal automata is undecidable when several suffixes are allowed [5].

We will now see that in the particular context of Theorem 5.4, time-abstract bisimulation is fine enough to solve control problems.

Lemma 5.6. *Let $\mathcal{A}$ be an o-minimal game, $\mathcal{P}$ a partition inducing a time-abstract bisimulation. If there exists a unique suffix on $\mathcal{P}$ associated with each $(q, y) \in Q \times V_2$ then $\mathcal{P} = \mathsf{Suf}(\mathcal{P})$.*

Applying Theorem 5.4 and Lemma 5.6 we get that $\mathsf{Suf}(\mathcal{P}_\mathcal{A})$ is suffix-stable, and using Proposition 4.4 we obtain that it is stable under π:

Proposition 5.7. *Let $\mathcal{A}$ be an o-minimal game. If there exists a unique suffix on $\mathcal{P}_\mathcal{A}$ associated with each $(q, y) \in Q \times V_2$ then $\mathsf{Suf}(\mathcal{P}_\mathcal{A})$ is finite and stable under the action of π.*

[7] This is is a particular case of reset for $\mathcal{M}$-game where we consider only constant functions for resets.

5.2 Synthesis of winning strategies

We now prove that under the hypotheses of Theorem 5.4, we can construct a *definable* strategy for the winning states. The effectiveness of this construction will be discussed in subsection 5.3.

Theorem 5.8. *Let $\mathcal{A}$ be an o-minimal game. If there exists a unique suffix associated with each $y \in V_2$ on $\mathcal{P}_{\mathcal{A}}$, then there exists a definable winning strategy for each $y \in \pi^*(\mathsf{Goal})$.*

Proof (Sketch). The complete proof is done in the appendix.

Given $P \subseteq \pi^*(\mathsf{Goal})$, by Proposition 4.2 we know there exists a winning strategy on P. We now point out a defi nable memoryless winning strategy, ***i.e.*** we build a defi nable function $\lambda : \{(q, x, t, y) \mid \gamma_q(x, t) = y\} \to M^+ \times \Sigma_c$. We defi ne λ by induction on the number of iterations of π.

Suppose we have already built a defi nable strategy on $W = \bigcup_{0 \leqslant i \leqslant k} \pi^i(\mathsf{Goal})$, and let us now consider $\pi(W) \setminus W$.

By Proposition 5.7 we know that $\pi(W) \setminus W$ is a fi nite union of pieces of $\mathsf{Suf}(\mathcal{P})$. Let P be one of these pieces. Let w be the unique suffi x associated with all states (q, y) of P. One can see that there exists a piece P' of $\mathsf{Suf}(\mathcal{P})$ and an action c in Σ_c such that $P' \xrightarrow{c} P''$ with $P'' \subseteq W$. The strategy in piece P will then be to perform action c after some delay. We now explain how to choose this delay)to reach piece P' of the partition).

Let (q, x, t, y) be such that $y \in P$ and $\gamma_q(x, t) = y$. Let us consider $\mathsf{Time}(x, t)$ the subset of M^+ defi ned as follows:

$$\mathsf{Time}(x, t) = \{t' \in M^+ \mid \gamma(x_0, t + t') \in P'\}.$$

This set is defi nable since P' is defi nable by Theorem 5.4.

By o-minimality we have that $\mathsf{Time}(x, t)$ is a fi nite union of points and open intervals. Let us denote by I the leftmost point or interval. Let us notice that I is defi nable. If I has a minimum m, we defi ne $\lambda(q, x, t, y) = (m, a)$. Otherwise two cases may occur. If I is bounded then it is of the form (m, m') or $(m, m']$ in this case we defi ne[8] $\lambda(q, x, t, y) = (\frac{1}{2}(m + m'), a)$. Finally if I has no minimum and is unbounded it is of the form (m, ∞) and in this case we defi ne $\lambda(q, x, t, y) = (m + 1, a)$. We summarize[9] the defi nition of λ on P as follows:

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

where $\varphi_1(x, t)$ is a formula which is true if and only if I (or $\mathsf{Time}(x, t)$) has a minimum and $\varphi_2(x, t)$ is a formula which is true if and only if I has no minimum and is bounded. Thus clearly λ is defi nable on P.

[8] Let us recall that every o-minimal ordered group is torsion free and divisible (see [19]), this implies there exists a unique y satisfying $y + y = (m + m')$, which we note $\frac{1}{2}(m + m')$.

[9] Let us notice that the way we extract a single point from $\mathsf{Time}(x, t)$ is nothing more than the *curve selection* for o-minimal expansions of ordered abelian groups, see [23, chap.6].

Since there are finitely many $P \in \mathsf{Suf}(\mathcal{P}_{\mathcal{A}})$ (see Theorem 5.4), we can conclude that λ is definable on $\pi^*(\mathsf{Goal})$. $\square$

Let us now illustrate Theorem 5.8 on two examples.

Example 5.9. Let us consider again the automaton shape of Example 2.9. We now define from $\mathcal{A}$ an o-minimal game $\mathcal{A}_s$ related to the spiral example (Example 3.4). The underlying o-minimal structure[10] $\mathcal{M}$ is $\langle \mathbb{R}, +, \cdot, 0, 1, <, \sin_{|[0,2\pi]}, \cos_{|[0,2\pi]} \rangle$. The o-minimal game $\mathcal{A}_s$ has the same set of locations, same Goal, same set of actions and same underlying finite automaton than $\mathcal{A}$ (*i.e.* Figure 1(a) represents also $\mathcal{A}_s$). The two differences between $\mathcal{A}$ and $\mathcal{A}_s$ are the guards and the continuous dynamics. Let us first define the guards. We have that g_B can be taken on B-states (*i.e.* points on the spiral) and g_C on C-states (points not on the spiral and different from the origin). The continuous dynamics in q_1 is the one described by the dynamical system of Example 3.4 (the continuous dynamics in q_2 and q_3 does not play any role). Clearly g_B, g_C and γ_{q_1} are definable in $\mathcal{M}$.

The winning strategy in point $(0,0)$ given by Theorem 5.8 is $\lambda(0,0,\theta,t) = (\frac{\theta}{2}, c)$ where c consists in taking the transition leading to state q_2 (which is winning).

Example 5.10. Let us notice that in the case of timed automata dynamics (described in Example 2.2), our definable strategies correspond in some sense to the realizable strategies obtained in [4].

5.3 Decidability result

Theorem 5.8 is an existential result. It claims that given an o-minimal game with suffix uniqueness hypothesis, there exists a definable strategy for each $y \in \pi^*(\mathsf{Goal})$, and by Theorem 5.4 we know that $\mathsf{Suf}(\mathcal{P})$ is finite. The conclusion of the previous subsection is that under hypothesis of Theorem 5.4 there exists a definable uniform memoryless winning strategy on each $P \in \mathsf{Suf}(\mathcal{P})$ such that $P \subseteq \pi^*(\mathsf{Goal})$.

We recall that a theory $\mathsf{Th}(\mathcal{M})$ is decidable iff there is an algorithm which can determine whether or not any sentence[11] 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 [7].

In general Theorem 5.8 does not allow to conclude that the control problem in an $\mathcal{M}$-structure is decidable. Indeed it depends on the decidability of $\mathsf{Th}(\mathcal{M})$.

However we can state the following theorem:

Theorem 5.11. *Let $\mathcal{M}$ be an o-minimal structure such that $\mathsf{Th}(\mathcal{M})$ is decidable and $\mathcal{C}$ the class of $\mathcal{M}$-automata $\mathcal{A}$ such that there exists a unique suffix on $\mathcal{P}_{\mathcal{A}}$ associated with each $(q,y) \in Q \times V_2$. Then the control problem in class $\mathcal{C}$ is decidable. Moreover if $\mathcal{A} \in \mathcal{C}$, the set of winning states $\pi^*(\mathsf{Goal})$ is computable and a strategy can be effectively computed for each $(q,y) \in \pi^*(\mathsf{Goal})$.*

[10] This structure is o-minimal (see [22]).
[11] *i.e.* a formula with no free variable.

Proof. By Theorem 5.4, for each $\mathcal{A} \in \mathcal{C}$, $\mathsf{Suf}(\mathcal{P}_{\mathcal{A}})$ is a definable finite partition respecting Goal; Corollary 5.7 ensures that this partition is stable under π. Hypothesis of Corollary 4.3 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.

The proof of Theorem 5.8 gives a way to compute a winning strategy. □

Remark 5.12. Let us notice that $\langle \mathbb{R}, <, +, 0, 1 \rangle$ and $\langle \mathbb{R}, <, +, \cdot, 0, 1 \rangle$ are examples of o-minimal structures with decidable theory.

Remark 5.13. Let us notice that the "unique suffix" assumption of Theorem 5.11 encompasses the continuous behavior allowed in [18] (where the dynamics γ is the flow of a vector field that does not depend on the time, and is thus time-deterministic). More general systems can also be handled, for example the spiral dynamics (Example 5.9) which is an infinitely branching system with unique suffix.

Remark 5.14. Let us notice that given $\mathcal{A}$ an o-minimal $\mathcal{M}$-game such that $\mathsf{Th}(\mathcal{M})$ is decidable, we can effectively decide if there exists a unique suffix $\mathcal{P}_{\mathcal{A}}$ associated with each point $(q, y) \in Q \times V_2$.

Remark 5.15. In fact Theorem 5.11 can be proved for a wider class than o-minimal systems: the condition that every variable is reset on every transition is only used to get that $\mathcal{P}_{\mathcal{A}}$ is stable under the action of cPred and uPred; if this condition is satisfied (and the dynamic in every state is o-minimal) the resets can be arbitrary.

Timed automata can be treated in this framework. Theorem 5.11 thus provides in particular a way to compute winning strategies for timed games.

6 Conclusion

In this paper we have studied the control problem of hybrid systems with general dynamics. We have shown that time-abstract bisimulation is not fine enough to solve them, which is a major difference with the discrete case. Using an encoding of trajectories by words [6], we have proved that the so-called suffix partition is a good abstraction for control. We have finally provided decidability and computability results for o-minimal hybrid systems. Our technique applies to timed automata, and we get the decidability of timed games [3], as well as the construction of winning strategies [4] as side results.

There are several interesting further research directions: we could try to relax the suffix uniqueness hypothesis, or assume only partial observability of the system.

References

1. R. Alur and D. Dill. A theory of timed automata. *Theoretical Computer Science*, 126(2):183–235, 1994.
2. R. Alur, T. A. Henzinger, G. Lafferriere, and G. J. Pappas. Discrete abstractions of hybrid systems. *Proc. of the IEEE*, 88:971–984, 2000.
3. E. Asarin, O. Maler, A. Pnueli, and J. Sifakis. Controller synthesis for timed automata. In *Proc. IFAC Symposium on System Structure and Control*, pages 469–474. Elsevier Science, 1998.

4. P. Bouyer, F. Cassez, E. Fleury, and K. 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.
5. T. Brihaye. Undecidability of the reachability problem for o-minimal hybrid systems, 2005. Submitted.
6. T. Brihaye. Words and bisimulation of dynamical systems. *Journal of Automata, Languages and Combinatorics*, 2005. To appear.
7. T. Brihaye and C. Michaux. On the expressiveness and decidability of o-minimal hybrid systems. *Journal of Complexity*, 21(4):447–478, 2005.
8. T. Brihaye, C. Michaux, C. Rivière, and C. 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.
9. V. Bruyère and O. Carton. Automata on linear orderings. In *Proc. 26th International Symposium on Mathematical Foundations of Computer Science (MFCS'01)*, volume 2136 of *Lecture Notes in Computer Science*, pages 236–247. Springer, 2001.
10. J. M. Davoren. Topologies, continuity and bisimulations. *Informatique Théorique et Applications*, 33(4-5):357–382, 1999.
11. L. de Alfaro, T. A. Henzinger, and R. 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.
12. E. Grädel, W. Thomas, and T. Wilke, editors. *Automata, Logics, and Infi nite Games: A Guide to Current Research*, volume 2500 of *Lecture Notes in Computer Science*. Springer, 2002.
13. T. 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.
14. T. 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.
15. T. A. Henzinger, B. Horowitz, and R. 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.
16. W. Hodges. *A Shorter Model Theory*. Cambridge University Press, 1997.
17. M. V. Korovina and N. 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.
18. G. Lafferriere, G. J. Pappas, and S. Sastry. O-minimal hybrid systems. *Mathematics of Control, Signals, and Systems*, 13(1):1–21, 2000.
19. A. Pillay and C. Steinhorn. Definable sets in ordered structures. *Transactions of the American Mathematical Society*, 295(2):565–592, 1986.
20. A. Rabinovitch. Automata over continuous time. *Theoretical Computer Science*, 300(1-3):331–363, 2003.
21. J. K. Truss. Infinite permutation groups ii. subgroups of small index. *Journal of Algebra*, 120(2):494–515, 1989.
22. L. Van den Dries. O-minimal structures. In *Logic: From Foundations to Applications*, Oxford Science Publications, pages 137–185. Oxford University Press, 1996.
23. L. van den Dries. *Tame Topology and O-Minimal Structures*, volume 248 of *London Mathematical Society Lecture Note Series*. Cambridge University Press, 1998.

A Proof of Proposition 4.2

Proposition 4.2. *Let $\mathcal{A} = (\mathcal{M}, Q, \mathsf{Goal}, \Sigma, \delta, \gamma)$ be an $\mathcal{M}$-game, and $(q, y) \in Q \times V_2$. Then, $(q, y) \in \pi^*(\mathsf{Goal})$ iff there is a winning strategy in $\mathcal{A}$ from (q, y).*

Proof. We first prove that if $(q, y) \in \pi^*(\mathsf{Goal})$ then there exists a winning strategy from (q, y). To this aim, we define a state-based winning strategy from any $(q, y) \in \pi^*(\mathsf{Goal})$ (a strategy is state-based when its value on an execution only depends on the last configuration of this execution). By notation misuse, we define the strategy λ on states (q, x, t, y) instead of executions.

We define a strategy λ on all sets $\bigcup_{0 \leqslant i \leqslant k} \pi^i(\mathsf{Goal})$ by induction on k, and prove that it is a winning strategy. If $k = 0$, we define λ to be any controllable action looping on Goal; it is winning by definition.

Suppose now that λ is already defined on $W = \bigcup_{0 \leqslant i \leqslant k} \pi^i(\mathsf{Goal})$ and is winning on these states. We now define λ on $\pi(W)$. Let $(q, x, t, y) \in Q \times V_1 \times M^+ \times V_2$: if $(q, y) \in W$, λ is already defined; if $(q, y) \in \pi(W) \setminus W$, then we know that $(q, y) \in \mathsf{Pred}_t\left(\mathsf{cPred}(W), \mathsf{uPred}(\overline{W})\right)$. For every (x, t) such that $\gamma_q(x, t) = y$, there exists $t' \in M^+$ and $c \in \Sigma_c$ with (t', c) enabled in (q, y) and $(q, y) \xrightarrow{t',c}_{x,t} (q', y')$ implies $(q', y') \in W$ and $\mathsf{Post}^{q,x}_{[t,t+t']}(W) \subseteq \overline{\mathsf{uPred}(\overline{W})}$. We set $\lambda(q, x, t, y) = (t', c)$ and show that this is a winning choice.

Let $\rho = (q, x, t, y) \xrightarrow{t_1,a_1} (q_1, x_1, t_1, y_1) \xrightarrow{t_2,a_2} \ldots$ be an execution compatible with λ. We have that either $t_1 = t'$ and $a_1 = c$, in which case $(q_1, y_1) \in W$, or $t_1 \leqslant t'$ and $a_1 \in \Sigma_u$, in which case $(q, y) \xrightarrow{t_1}_{x,t} (q', y') \xrightarrow{a_1} (q_1, y_1)$ 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.

We now show that if there exists a strategy λ winning from (q, y) then $(q, y) \in \pi^*(\mathsf{Goal})$. Set $W = \pi^*(\mathsf{Goal})$, by contradiction suppose that $(q, y) \notin W$, we will construct a non-winning execution compatible with λ. As $(q, y) \notin W$, there exists $(x, t) \in V_1 \times M^+$ such that $\gamma_q(x, t) = y$, and for all $t' \in M^+$, $(q, y) \rightarrow^{t'}_{x,t} (q', y')$ implies $(q', y') \notin \mathsf{cPred}(W)$ or $\mathsf{Post}^{q,x}_{[t,t+t']}(W) \cap \mathsf{uPred}(\overline{W}) \neq \varnothing$. Fix such an $(x, t) \in V_1 \times M^+$, and set $(t', c) = \lambda(q, x, t, y)$.

There exists (q_1, x_1, t_1, y_1) with $(q_1, y_1) \notin W$ such that either $(q, x, t, y) \xrightarrow{t',c} (q_1, x_1, t_1, y_1)$ or there exists $t'' \leqslant t'$ and $u \in \Sigma_u$ with $(q, x, t, y) \xrightarrow{t'',u} (q_1, x_1, t_1, y_1)$. 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 $\rho = (q, x, t, y) \xrightarrow{t_1,a_1} (q_1, x_1, t_1, y_1) \xrightarrow{t_2,a_2} \ldots$ 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. □

B Proof of Corollary 4.3

Corollary 4.3. *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[12]*, and stable by π. Then the control problem in the class $\mathcal{C}$ is decidable. Moreover if $\mathcal{A} \in \mathcal{C}$, the set of winning states of $\mathcal{A}$ is computable.*

Proof. Let $\mathcal{M}$ be a structure and $\mathcal{C}$ a class of automata satisfying the hypotheses and take $\mathcal{A} \in \mathcal{C}$.

By proposition 4.2 the set of winning states is $\pi^*(\mathsf{Goal})$. As $\mathcal{P}$ is stable under π, $\pi^*(\mathsf{Goal})$ is a finite union of pieces of $\mathcal{P}$. Hence there exists $n \in \mathbb{N}$ such that $\pi^*(\mathsf{Goal}) = \pi^n(\mathsf{Goal})$. As π and Goal are definable, we have that $\pi^i(\mathsf{Goal})$ is definable and as $\mathsf{Th}(\mathcal{M})$ is decidable we can test if $\pi^i(\mathsf{Goal}) = \pi^{i+1}(\mathsf{Goal})$, we can thus effectively find a representation of $\pi^*(\mathsf{Goal})$.

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

C Proof of Lemma 5.6

Lemma 5.6. *Let $\mathcal{A}$ be an o-minimal game, $\mathcal{P}$ a partition inducing a time-abstract bisimulation. If there exists a unique suffix on $\mathcal{P}$ associated with each $(q, y) \in Q \times V_2$ then $\mathcal{P} = \mathsf{Suf}(\mathcal{P})$.*

Proof. We work in a given location $q \in Q$ and for convenience we denote by y the state (q, y). We also use the shortcut $y \rightarrow_\gamma y'$ for $\exists\, t \geqslant 0\; y \xrightarrow{t}_\gamma y'$.

Let $y_1, y_1' \in A_1$ for some $A_1 \in \mathcal{P}$. We will prove that $\mathsf{Suf}_\mathcal{P}(y_1) = \mathsf{Suf}_\mathcal{P}(y_1')$.

In the context of o-minimal systems, the suffix associated to a point is a finite word [7], so let $\omega_1 = A_1 ... A_n$ be the unique suffix associated with y_1, we can build the following sequence of transitions.

$$y_1 \rightarrow_\gamma y_2 \rightarrow_\gamma ... \rightarrow_\gamma y_n,$$

with $y_i \in A_i$ for $i = 1, ..., n$. Since y_1 and y_1' are time-abstract bisimilar,

we can build a similar sequence of transitions.

$$y_1' \rightarrow_\gamma y_2' \rightarrow_\gamma ... \rightarrow_\gamma y_n',$$

with $y_i' \in A_i$ for $i = 1, ..., n$.

Let us now prove that the suffix uniqueness hypothesis implies that there exists $x \in V_1$ and t_1,...,$t_n \in M$ with $t_1 \leqslant ... \leqslant t_n$ such that $\gamma(x, t_1) = y_1'$, and $\gamma(x, t_i) \in A_i$ for $i = 1, ..., n$; meaning that ω_1 is a sub-word of ω_1' (the unique suffix associated with y_1'). Clearly we can find x, t_1, t_2 with $t_1 \leqslant t_2$, $y_1' = \gamma(x, t_1) \in A_1$ and $\gamma(x, t_2) \in A_2$ (since $y_1' \rightarrow_\gamma y_2'$). Let us suppose, for a contradiction, that given x, t_1, t_2 such that $t_1 \leqslant t_2$, $\gamma(x, t_1) \in A_1$ and $\gamma(x, t_2) \in A_2$ we have that $\gamma(x, t_3) \notin A_3$ for all $t_3 \geqslant t_2$. In particular using the suffix uniqueness hypothesis, this means that the unique suffix associated with y_2' does not contain the letter A_3. This contradicts the existence of the transition $y_2' \rightarrow_\gamma y_3'$ where $y_3' \in A_3$. Thus we can find t_3 with the desired conditions. Iterating the same argument we find the other t_i's.

Similarly, we can prove that ω_1' is a sub-word of ω_1. Hence $\omega_1 = \omega_1'$ since they are finite words by o-minimality assumptions. Thus $\mathsf{Suf}_\mathcal{P}(y_1) = \mathsf{Suf}_\mathcal{P}(y_1')$. □

[12] *i.e.* Goal is a union of pieces of $\mathcal{P}$

D Proof of Theorem 5.8

In order to prove Theorem 5.8, we have to associate a couple (t, a) with each state $q \in \pi^*(Goal)$. We proceed in two steps. First we prove that we can define a winning strategy uniform w.r.t. the controlled actions on each piece P of $\mathcal{P}$. Then we show how to pick a time in a definable way. We start by defining what we mean by uniform strategy.

Definition D.1. Given $\mathcal{A}$ an $\mathcal{M}$-game, $a \in \Sigma_c$, $\mathcal{P}$ a partition of $Q \times V_2$ and $P \in \mathcal{P}$. We say that a strategy λ is *a-uniform on P* if the following condition holds:

$$\begin{aligned}&\forall (q, x, t) \in Q \times V_1 \times M^+\\ &\quad \big(q, \gamma(x, t)\big) \in P \;\Rightarrow\; \exists t' \in M^+ \text{ s.t. } \lambda\big(q, x, t, \gamma(x, t)\big) = (t', a)\end{aligned}$$

We say that a strategy λ is *uniform* if for all $P \in \mathcal{P}$, there exists $a \in \Sigma_c$ such that λ is *a-uniform on P*.

Lemma D.2. *Under the hypotheses of Theorem 5.4, there exists a uniform winning strategy on each piece of $P \in \mathsf{Suf}(\mathcal{P}_\mathcal{A})$ such that $P \subseteq \pi^*(\mathsf{Goal})$.*

Proof. Given $y \in \pi^*(\mathsf{Goal})$, by Proposition 4.2 we know that there exists a winning strategy from y. We now have to point out a uniform winning strategy. Following the lines of the proof of Proposition 4.2, we build the definable strategy by induction on the number of iterations of π. Let us suppose we already built a uniform strategy on each piece of W, let us now consider $\pi(W) \setminus W$ (where $W = \cup_{0 \leqslant i \leqslant k} \pi^i(\mathsf{Goal})$).

By Corollary 5.7 we know that $\pi(W) \setminus W$ is a union of pieces of $\mathsf{Suf}(\mathcal{P}_\mathcal{A})$. Let P be one of these pieces. Given (q, x_0, t_0) such that $\gamma(x_0, t_0) = y_0 \in P$, by Proposition 4.2 we know there exists $(t'_0, a) \in \Sigma_c \times M^+$ such that defining $\lambda(q, x_0, t_0, y_0)$ by (t'_0, a) will make λ a winning strategy on (q, y_0). We now prove that given any (q, x, t) such that $\gamma(x, t) = y \in P$ there exists $t' \in M^+$ such that defining $\lambda(q, x, t, y)$ to be (t', a) will make λ a winning strategy on (q, y).

The previous statement holds by the suffix uniqueness hypothesis. Indeed given given any (q, x, t) such that $\gamma(x, t) = y \in P$, there exists $t' \in M^+$ such that $\omega_{(x,t,t')} = \omega_{(x_0,t_0,t'_0)}$ (for the dynamics γ_q). By choosing this t' we obtain the desired result. In particular, with this choice, there exists a piece $P' \in \mathsf{Suf}(\mathcal{P}_\mathcal{A})$ such that for all (q, x, t) such that $\gamma(x, t) = y \in P$ we have that $\big(q, \gamma(x, t + t')\big) \in P'$. □

Theorem 5.8. *Let $\mathcal{A}$ be an o-minimal game. If there exists a unique suffix associated with each $y \in V_2$ on $\mathcal{P}_\mathcal{A}$, then there exists a definable winning strategy for each $y \in \pi^*(\mathsf{Goal})$.*

Proof. Given $P \subseteq \pi^*(\mathsf{Goal})$, by Proposition 4.2 we know there exists a winning strategy on P. We now point out a definable memoryless winning strategy, *i.e.* we build a definable function $\lambda : \{(q, x, t, y) \mid \gamma_q(x, t) = y\} \to M^+ \times \Sigma_c$. Again following the lines of the proof of Proposition 4.2, we define λ by induction on the number of iterations of π.

Suppose we have already built a definable strategy on $W = \bigcup_{0 \leqslant i \leqslant k} \pi^i(\mathsf{Goal})$, and let us now consider $\pi(W) \setminus W$.

By Corollary 5.7 we know that $\pi(W) \setminus W$ is a finite union of pieces of $\mathsf{Suf}(\mathcal{P})$. Let P be one of these pieces. By Lemma D.2 there exists an a-uniform winning strategy on P for some $a \in \Sigma_c$. Let P' be the piece of $\mathsf{Suf}(\mathcal{P})$ appearing at the end of the proof of Lemma D.2.

Given (q, x, t, y) such that $y \in P$ and $\gamma_q(x, t) = y$, let us consider $\mathsf{Time}(x, t)$ the subset of M^+ defined as follows:

$$\mathsf{Time}(x,t) = \{t' \in M^+ \mid \gamma(x_0, t + t') \in P'\}.$$

This set is definable since P' is definable by Theorem 5.4.

By o-minimality we have that $\mathsf{Time}(x, t)$ 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 $\lambda(q, x, t, y) = (m, a)$. 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 $\lambda(q, x, t, y) = (\frac{m+m'}{2}, a)$. Finally if I has no minimum and is unbounded it is of the form (m, ∞) and in this case we define $\lambda(q, x, t, y) = (m + 1, a)$. We summarize the definition of λ on P as follows:

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

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

Since there are finitely many $P \in \mathsf{Suf}(\mathcal{P}_{\mathcal{A}})$ (see Theorem 5.4), we can conclude that λ is definable on $\pi^*(\mathsf{Goal})$. $\square$