A

Automatic Synthesis of Switching Controllers for Linear Hybrid Systems: Reachability Control

Massimo Benerecetti, University of Naples "Federico II", Italy
Marco Faella, University of Naples "Federico II", Italy

We consider the problem of computing the controllable region of a Linear Hybrid Automaton with controllable and uncontrollable transitions, w.r.t. a reachability objective. We provide an algorithm for the finite-horizon version of the problem, based on computing the set of states that must reach a given non-convex polyhedron while avoiding another one, subject to a polyhedral constraint on the slope of the trajectory. Experimental results are presented, based on an implementation of the proposed algorithm on top of the tool SpaceEx.

Additional Key Words and Phrases: hybrid automata, controller synthesis, reachability games

1. INTRODUCTION

Hybrid systems are a type of non-linear dynamic systems, characterized by the presence of continuous and discrete variables. Hybrid automata [Henzinger 1996] are the most common syntactic variety of hybrid system: a finite set of locations, similar to the states of a finite automaton, represents the value of the discrete variables. The current location, together with the current value of the (continuous) variables, form the instantaneous description of the system. Change of location happens via discrete transitions, and the evolution of the variables is governed by differential inclusions attached to each location. In a Linear Hybrid Automaton (LHA), the allowed differential inclusions are of the type $\dot{x} \in P$, where $\dot{x}$ is the vector of the first derivatives of all variables and $P \subseteq \mathbb{R}^n$ is a convex polyhedron. Notice that differential inclusions are non-deterministic, allowing for infinitely many solutions with the same starting conditions.

We study LHAs whose discrete transitions are partitioned into controllable and uncontrollable ones, giving rise to a 2-player model called Linear Hybrid Game (LHG): on one side the controller, which can only issue controllable transitions; on the other side the environment, which can choose the trajectory of the variables and can take uncontrollable transitions whenever they are enabled.

As control goal, we consider reachability, i.e., the objective of reaching a given set of target states. As we show in Section 2.2, it is easy to show that the reachability control problem is undecidable, being harder than the standard reachability verification (i.e., 1-player reachability) for LHAs [Henzinger et al. 1998]. We present the first exact algorithm for 1-step controllability under a reachability objective, namely reaching the target region with at most one discrete transition. In turn, this provides an exact algorithm for bounded-horizon reachability control (i.e., reaching the target within a fixed number of discrete steps), and a semi-algorithm[1] for the infinite-horizon case.

We recently presented a semi-algorithm for the control problem of LHGs with *safety* objectives [Benerecetti et al. 2011a; 2013]. Although the control goal we examine, as a language of infinite traces, is the dual of safety, the corresponding synthesis problem is not, because our game model is asymmetric (continuous-time trajectories are always uncontrollable). Hence, it is not possible to solve the control problem with reachability goal T by exchanging the roles of the two players and then solving the safety control problem with goal $\overline{T}$ (i.e., the complement of T).

[1]In other words, a procedure that may or may not terminate, and that provides the correct answer whenever it terminates.

On the one hand, the one-step safety control problem can be solved by computing the may-reach-while-avoiding operator RWA^{m}. Given two sets of states U and V, $RWA^{\mathrm{m}}(U,V)$ collects the states from which there exists a system trajectory that reaches U while avoiding V at all times. On the other hand, the one-step *reachability* control problem requires a different operator, called must-reach-while-avoiding and denoted by $RWA^{\mathrm{M}}(U,V)$. As suggested by its name, such operator computes the set of states from which *all* system trajectories reach U while avoiding V. The main technical result of this paper is that the first operator can be used to compute the latter, once a suitable over-approximation of $RWA^{\mathrm{M}}(U,V)$ is available (Theorem 4.5). Moreover, we present two effective ways to obtain such an over-approximation, which are compared experimentally in Section 7.

To the best of our knowledge, the reachability control goal was never considered for LHGs. This paper extends the results presented in [Benerecetti et al. 2012], where the problem was first considered. Compared to the preliminary version, here we present a second over-approximation, which significantly improves the performance of the algorithm in several cases. Moreover, the experiments, based on the tool SpaceEx, are entirely new and compare the performance of the two over-approximations proposed in the paper. Finally, several proofs have been extended with further details and the decidability status of the problem restricted to non-Zeno systems has been addressed in a new subsection.

Related work. The idea of automatically synthesizing controllers for dynamic systems arose in connection with discrete systems [Ramadge and Wonham 1987]. Then, the same idea was applied to real-time systems modeled by timed automata [Maler et al. 1995], thus coming one step closer to the continuous systems that control theory usually deals with. Finally, it was the turn of hybrid systems [Henzinger et al. 1999; de Alfaro et al. 2001], and in particular of Linear Hybrid Automata [Wong-Toi 1997], the very model that we analyze in this paper. Wong-Toi proposed a symbolic semi-algorithm to compute the controllable region of a LHA w.r.t. a safety goal [Wong-Toi 1997].

Tomlin et al. [2000], Lygeros et al. [1999] and Balluchi et al. [2003] analyze much more expressive models, with generality in mind rather than automatic synthesis. Asarin et al. [2000] investigate the synthesis problem for hybrid systems where all discrete transitions are controllable and the trajectories satisfy given linear differential equations of the type $\dot{\mathrm{x}} = A\mathrm{x}$. The expressive power of these constraints is incomparable with the one offered by the differential inclusions occurring in LHAs and LHGs. In particular, linear differential equations give rise to deterministic trajectories, while differential inclusions are non-deterministic. In control theory terms, differential inclusions can represent the presence of environmental *disturbances*. Bouyer et al. [2010] propose a general abstraction technique for hybrid systems, and focus on decidable classes of o-minimal automata.

A series of papers deals with the synthesis of continuous feedback control laws with the objective of reaching a given facet of a simplex [Habets et al. 2006; Lin and Broucke 2006]. Compared to LHGs, the systems of interest have a more expressive dynamics, but they allow no disturbances, so they are deterministic once control is applied.

Structure of the paper. The rest of the paper is organized as follows. Section 2 introduces and motivates the model. The proposed semi-algorithm is presented as divided in two layers: Section 3 illustrates the outer layer, dealing with multiple locations and discrete transitions, while Section 4 focuses on the geometric problem arising from the analysis of a single location. In particular, it introduces the operator RWA^{M} and shows how to compute it by applying RWA^{m} to suitable over-approximations of the desired result. Sections 5 and 6 focus on the computation of such over-approximations using simple geometric operations on polyhedra. Section 7 reports some experiments performed on our implementation of the procedure in SpaceEx [Frehse et al. 2011]. Finally, some conclusions are drawn in Section 8.

2. LINEAR HYBRID GAMES

A *convex polyhedron* is a subset of $\mathbb{R}^n$ that is the intersection of a finite number of strict and non-strict affine half-spaces. A *polyhedron* is a subset of $\mathbb{R}^n$ that is the union of a finite number of convex polyhedra. For a general (i.e., not necessarily convex) polyhedron $G \subseteq \mathbb{R}^n$, we denote by $cl(G)$ its topological closure, and by $[\![G]\!] \subseteq 2^{\mathbb{R}^n}$ its representation as a finite set of convex polyhedra.

Given an ordered set $X = \{x_1, \ldots, x_n\}$ of variables, a *valuation* is a function $v : X \to \mathbb{R}$. Let $Val(X)$ denote the set of valuations over X. There is an obvious bijection between $Val(X)$ and $\mathbb{R}^n$, allowing us to extend the notion of (convex) polyhedron to sets of valuations. We denote by $CPoly(X)$ (resp., $Poly(X)$) the set of convex polyhedra (resp., polyhedra) on X. Let A be a set of valuations, states or points in $\mathbb{R}^n$, we denote by $\overline{A}$ its complement.

We use $\dot{X}$ to denote the set $\{\dot{x}_1, \ldots, \dot{x}_n\}$ of dotted variables, used to represent the first derivatives, and X' to denote the set $\{x'_1, \ldots, x'_n\}$ of primed variables, used to represent the new values of variables after a transition. Given two valuations $u, v \in Val(X)$, we denote by $u \otimes v$ the valuation in $Val(X \cup X')$ obtained by conjoining u and v while renaming the variables of v. Conversely, for $w \in Val(X \cup X')$, we denote by $w\lfloor_X$ and $w\lfloor_{X'}$ the projections of w on X and X', respectively. Notice that $w = w\lfloor_X \otimes w\lfloor_{X'}$. Arithmetic operations on valuations are defined in the straightforward way.

A *trajectory* over X is a function $f : \mathbb{R}^{\geq 0} \to Val(X)$ that is differentiable but for a finite subset of $\mathbb{R}^{\geq 0}$. The issues arising from the stronger requirement of differentiability in every time point have been investigated elsewhere [Benerecetti and Faella 2013] and are out of the scope of this paper. Let $Trj(X)$ denote the set of trajectories over X. The *derivative* $\dot{f}$ of a trajectory f is defined in the standard way and it is a trajectory over $\dot{X}$.

Definition 2.1. A *Linear Hybrid Game (LHG)* $(Loc, X, Edg_c, Edg_u, Flow, Inv, Init)$ consists of the following components:

— A finite set Loc of *locations*.
— A finite set $X = \{x_1, \ldots, x_n\}$ of real-valued *variables*.
— Two sets $Edg_c, Edg_u \subseteq Loc \times Poly(X \cup X') \times Loc$ of *controllable* and *uncontrollable transitions*, respectively.
— A mapping $Flow : Loc \to CPoly(\dot{X})$, called the *flow constraint*.
— A mapping $Inv : Loc \to Poly(X)$, called the *invariant*.
— A mapping $Init : Loc \to Poly(X)$, contained in the invariant, defining the *initial states* of the automaton.

A *state* is a pair $\langle l, v \rangle$ of a location l and a valuation $v \in Val(X)$. Transitions describe instantaneous changes of location, in the course of which the variables may change their value. Each transition $(l, J, l') \in Edg_c \cup Edg_u$ consists of a *source location* l, a *target location* l', and a *jump relation* $J \in Poly(X \cup X')$, that specifies how the variables may change their value during the transition. The projection of J on X contains the valuations for which the transition is enabled, a.k.a. a *guard*. Jump relations generalize assignments to Boolean combinations of linear inequalities over current and next-state variable valuations. The flow constraint attributes to each location a set of valuations over the first derivatives of the variables, which determines how variables can change over time.

Intuitively, our objective is to synthesize a control strategy that exploits controllable transitions to achieve a reachability goal, regardless of the continuous-time evolution and uncontrollable transitions, both governed by the environment.

We use the abbreviations $S = Loc \times Val(X)$ for the set of states and $Edg = Edg_c \cup Edg_u$ for the set of all transitions. Moreover, we let $InvS = \bigcup_{l \in Loc} \{l\} \times Inv(l)$ and $InitS = \bigcup_{l \in Loc} \{l\} \times Init(l)$. Notice that $InvS$ and $InitS$ are sets of states. Given a set of states A and a location l, we denote by $A\downharpoonright_l$ the projection of A on l, i.e. $\{v \in Val(X) \mid \langle l, v \rangle \in A\}$.

2.1. Semantics

The behavior of an LHG is based on two types of transitions: *discrete* transitions correspond to the Edg component, and produce an instantaneous change in both the location and the variable valuation; *timed* transitions describe the change of the variables over time in accordance with the $Flow$ component.

Given a set $F \subseteq \mathbb{R}^n$, a trajectory $f \in Trj(X)$ is called *admissible w.r.t. F* if, whenever $\dot{f}(\delta)$ is defined, we have that $\dot{f}(\delta) \in F$. We denote by $Adm(u, F)$ the set of trajectories that start from $u \in \mathbb{R}^n$ and are admissible w.r.t. F. Given a state $s = \langle l, v \rangle$, let $loc(s) = l$ and $val(s) = v$. With a slight abuse of notation, we write $Adm(s)$ for $Adm(v, Flow(l))$. Additionally, for $f \in Adm(s)$, the ***span*** of f in l, denoted by $span(f, l)$ is the set of all values $\delta \geq 0$ such that $\langle l, f(\delta') \rangle \in InvS$ for all $0 \leq \delta' \leq \delta$. Intuitively, δ is in the span of f iff f never leaves the invariant in the first δ time units. If all non-negative reals belong to $span(f, l)$, we write $\infty \in span(f, l)$.

Runs. Given states s, s', and a transition $e \in Edg$, there is a ***discrete step*** $s \xrightarrow{e} s'$ with ***source*** s and ***target*** s' iff: *(i)* $s, s' \in InvS$, *(ii)* $e = (loc(s), J, loc(s'))$, and *(iii)* $val(s) \otimes val(s') \in J$. Whenever there is a discrete step $s \xrightarrow{e} s'$, we say that e is ***enabled in*** s.

There is a ***timed step*** $s \xrightarrow{f,\delta} s'$ with ***duration*** $\delta \in \mathbb{R}^{\geq 0}$ and trajectory $f \in Adm(s)$ iff: *(i)* $s \in InvS$, *(ii)* $\delta \in span(f, loc(s))$, and *(iii)* $s' = \langle loc(s), f(\delta) \rangle$. For technical convenience, we admit timed steps of duration zero[2]. The special timed step denoted by $s \xrightarrow{f,\infty}$ represents the case when the system follows a trajectory forever. This is only allowed if $\infty \in span(f, loc(s))$. A ***joint step*** $s \xrightarrow{f,\delta,e} s'$ represents the timed step $s \xrightarrow{f,\delta} \langle loc(s), f(\delta) \rangle$ followed by the discrete step $\langle loc(s), f(\delta) \rangle \xrightarrow{e} s'$. Finally, a ***run*** is a sequence

$$r = s_0 \xrightarrow{f_0,\delta_0} s'_0 \xrightarrow{e_0} s_1 \xrightarrow{f_1,\delta_1} s'_1 \xrightarrow{e_1} s_2 \cdots s_n \cdots \tag{1}$$

of alternating timed and discrete steps, such that either the sequence is infinite, or it ends with a timed transition of the type $s_n \xrightarrow{f,\infty}$. If the number of steps in r is finite, we define $len(r) = n$ to be the length of the run, otherwise we set $len(r) = \infty$. The above run is ***non-Zeno*** if for all $\delta \geq 0$ there exists $i \geq 0$ such that $\sum_{j=0}^{i} \delta_j > \delta$. We denote by $States(r)$ the set of all states visited by r. Formally, $States(r)$ is the set of all states $\langle loc(s_i), f_i(\delta) \rangle$, for all $0 \leq i \leq len(r)$ and all $0 \leq \delta \leq \delta_i$. Notice that the states from which discrete transitions start (states s'_i in (1)) appear in $States(r)$. Moreover, if r contains a sequence of one or more zero-time timed transitions, all intervening states appear in $States(r)$.

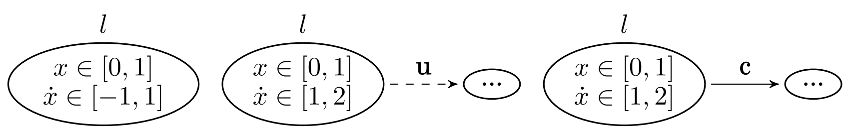
For a set of states A and a transition $e \in Edg$, let $Pre(e, A)$ be the set of states in $InvS$ where transition e is enabled and may lead to A. For $x \in \{u, c\}$, let $Pre_x(A) = \bigcup_{e \in Edg_x} Pre(e, A)$.

[2]Timed steps of duration zero can be disabled by adding a clock variable t to the automaton and requesting that each discrete transition happens when $t > 0$ and resets t to 0 when taken.

Zenoness and well-formedness. A well-known problem of real-time and hybrid systems is that definitions like the above admit runs that take infinitely many discrete transitions in a finite amount of time (i.e., *Zeno* runs), even if such behaviors are physically meaningless. In this paper, we assume that the hybrid automaton under consideration generates no such runs. This can be achieved using different syntactical constraints. For instance, one can use an extra variable, representing a clock, to ensure that the delay between any two location switches is bounded from below by a constant, called *minimum dwell time*. We leave it to future work to combine our results with the more sophisticated approaches to Zenoness known in the literature [Balluchi et al. 2003; de Alfaro et al. 2003].

Moreover, we assume that the hybrid automaton under consideration is *non-blocking*, i.e., before all system trajectories leave the invariant, there must be an uncontrollable transition enabled. Formally, for all states s in the invariant, if all trajectories $f \in Adm(s)$ eventually leave the invariant, there exists one such trajectory f and a time $\delta \in span(f, loc(s))$ such that $s' = \langle loc(s), f(\delta) \rangle$ is in the invariant and there is an uncontrollable transition that is enabled in s'. If a hybrid automaton is non-Zeno and non-blocking, we say that it is *well-formed*. In the following, all hybrid automata are assumed to be well-formed.

Example 2.2. Consider the LHGs in Figure 1, in which locations contain the invariant (first line) and the flow constraint (second line). Solid (resp., dashed) edges represent controllable (resp., uncontrollable) transitions, and guards are `true`. The fragment in Figure 1(a) is non-blocking, because the system may choose derivative $\dot{x} = 0$ and remain indefinitely in location l. The fragment in Figure 1(b) is also non-blocking, because the system cannot remain in l forever, but an uncontrollable transition leading outside is always enabled. Finally, the fragment in Figure 1(c) is blocking, because the system cannot remain in l forever, and no uncontrollable transition is enabled.



(a) Non-blocking. (b) Non-blocking. (c) Blocking.

Fig. 1. Three LHG fragments.

The role of the invariant. In our model, the invariant is a physical constraint of the system and not a control objective. Accordingly, non-blocking systems are able to indefinitely evolve in time, with no controller intervention, while remaining in their invariant at all times. The invariant then serves two purposes:

(1) it constrains the continuous-time evolutions: legal trajectories simultaneously satisfy the differential inclusion and the invariant of the current location;
(2) it indirectly forces the occurrence of uncontrollable transitions: if all trajectories exit from the invariant, an uncontrollable transition is bound to happen.

Other researchers have adopted different interpretations of invariants. Asarin et al. [2000] identify the invariants with the (safety) control objective. Their model features deterministic trajectories (no disturbances) and does not support uncontrollable transitions. So, purposes 1 and 2 above are void and the invariant is free to serve as the control goal.

Henzinger et al. [1999] support both uncontrollable transitions and disturbances. They additionally distinguish between the invariant of the environment and the invariant of the controller, whereas the control goal is a separate notion. Our invariants serve the exact same purposes as their environment invariants.

The maze example. As a motivating example, consider a vehicle navigating a maze, by taking 90-degree left or right turns: the possible directions are North (N), South (S), West (W) and East (E). One time unit (say, second) must pass between two changes of direction, while the vehicle speed is 2 unit of length (say, meters) per second. The corridors are 1 meter wide and the goal consists in reaching a target area positioned along the corridors.

Figure 2 shows the sketch of an LHG modeling the system: we have one location for each direction, where the derivative of the position variables (x and y) are set according to the corresponding direction. The variable t represents a clock ($\dot{t} = 1$) that is used to enforce a one-time-unit delay between turns. Each change of direction is modeled by a controllable transition, enabled when $t \geq 1$. The invariant in all locations corresponds to the shape of the maze, as shown in Figure 10.

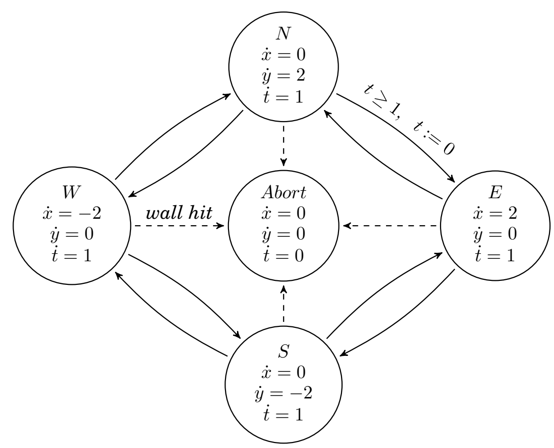


Fig. 2. The LHG for the maze example. Solid (resp., dashed) edges represent controllable (resp., uncontrollable) transitions.

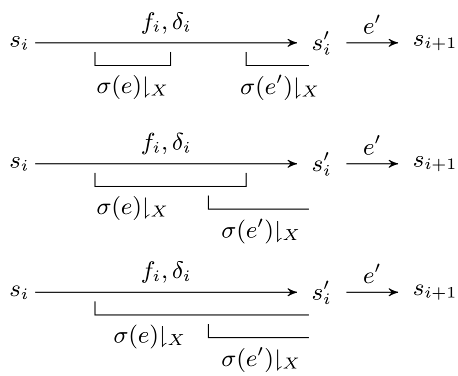
The sink location *Abort* models all the cases in which the vehicle hits one of the walls of the maze. Each wall is modeled by an uncontrollable transition, enabled when the vehicle hits the wall, leading to the *Abort* location. These uncontrollable transitions are not subject to any timing constraint, as opposed to the controlled ones. As a consequence, the resulting LHG does not satisfy the minimum dwell time constraint, since the location switch leading to the *Abort* location may occur within an arbitrary small delay since the last change of direction. The LHG is, however, non-Zeno, as no discrete transition allows the system to leave the *Abort* location, once reached.

Strategies. A *strategy* is a function $\sigma : Edg_c \to Poly(X, X')$ such that for all $e = (l, J, l') \in Edg_c$ we have that $\sigma(e) \subseteq J$. Strategies assign to each controllable transition a possibly non-convex polyhedron, which is contained in the jump relation of the transition. The intended meaning is that the strategy restricts controllable transitions so that they can be taken from a given subset of their original guard and they lead to a given subset of their original set of destinations. In particular, non-determinism in a controllable transition is resolved by the controller. This contrasts with other papers [Henzinger et al. 1999], in which non-determinism is always resolved in favor of the environment (i.e., *adversarial non-determinism*). If the latter semantics is desired, one can add for each controllable transition an intermediate location followed by an uncontrollable transition that will be responsible for resolving the non-deterministic choice. On the other hand, we conjecture that there is no uniform way to simulate our semantics using adversarial non-determinism.

We stipulate below that when the system enters the "activated" region $\sigma(e)\downarrow_X$ (i.e., the projection of $\sigma(e)$ on the variables X), some discrete transition (not necessarily e)

must be taken before the system exits from that region. Without this assumption, a strategy could only (de-)activate controllable transitions, but it would have no way of actually forcing a controllable transition to happen. For instance, the joint steps in Figures 3(a) and 3(b) are not valid because the system leaves the activated region of transition e before taking a discrete transition. The joint step in Figure 3(c) instead is valid.

In other works [Asarin et al. 2000], this effect is achieved by allowing the controller to also restrict the invariant. In our framework that approach would be incorrect, as a restricted invariant may force the environment to take more actions, possibly allowing the controller to unduly win the game.



(a) An illegal joint step.

(b) An illegal joint step.

(c) A legal joint step.

Fig. 3. The semantics of strategies.

Following the intuition above, we say that a run like (1) is *consistent* with a strategy σ if for all $0 \leq i < len(r)$ the following conditions hold:

— if $e_i \in Edg_c$ then $val(s'_i) \otimes val(s_{i+1}) \in \sigma(e_i)$;
— if $\delta_i < \infty$, then for all $e = (loc(s_i), J, l') \in Edg_c$ such that $f_i(\delta) \in \sigma(e)\lfloor_X$, for some $\delta \in [0, \delta_i]$, it holds that $f_i(\delta') \in \sigma(e)\lfloor_X$, for all $\delta' \in [\delta, \delta_i]$;
— if $\delta_i = \infty$ then for all $\delta \geq 0$ and all $e = (loc(s_i), J, l') \in Edg_c$, it holds that $f_i(\delta) \notin \sigma(e)\lfloor_X$.

The first condition ensures that if the i-th transition is controllable, then it is taken according to the prescriptions of the strategy. The second condition ensures that the system does not exit from an activated region without taking an action, while the third condition prevents the system to stay forever in an activated region without taking any action. We denote by $Runs(s, \sigma)$ the set of runs starting from the state s and consistent with the strategy σ.

Reachability control problem. Given a hybrid automaton and a set of states $T \subseteq InvS$, the *reachability control problem* asks whether there exists a strategy σ such that, for all initial states $s \in InitS$ and all runs $r \in Runs(s, \sigma)$ it holds $States(r) \cap T \neq \emptyset$. We call the above σ a *winning strategy*.

2.2. Undecidability

An LHG is *deterministic* if for all states s there exists a unique joint step starting from s. A deterministic LHG induces a single run, where all discrete steps are uncontrollable. Indeed, if a controllable transition occurred, there would be another run in which that transition would not be taken, contradicting the uniqueness of the run. As a consequence, removing all controllable transitions from a deterministic LHG leads to an equivalent LHG, which admits a single vacuous strategy (i.e., a function with an empty domain).

Recall that Henzinger et al. [1998], the authors prove that the reachability problem for a restricted class of LHAs is undecidable, by reducing the halting problem for deterministic 2-counter machines (2CMs), which is known to be undecidable [Minsky 1967]. It is easy to verify that the 4-variable LHA corresponding to a given 2CM is deterministic due to the following properties[3]: all flow constraints are of the type $\dot{x}_i = k_i \in \mathbb{R}_{>0}$, for all variables x_i; all guards contain a constraint of the type $x_i = c_i$, for some variable x_i and constant $c_i \in \mathbb{R}$; the jump relation of all transitions constrains each variable to either retain its old value or assume value zero; guards belonging to different transitions do not overlap. Moreover, said LHA satisfies the minimum dwell time property, since every transition leading from one location to another resets a clock a and is only enabled when $a = W$, for a positive constant W. We can then prove the following result.

THEOREM 2.3. *The reachability control problem is undecidable for deterministic LHGs with minimum dwell time.*

PROOF. We reduce the halting problem for 2CMs to the reachability control problem for deterministic LHGs with minimum dwell time. Given a 2CM, consider the corresponding LHA and its target region T as defined by Henzinger et al. [1998] and mentioned above. We trivially convert it into a determinisic LHG by stipulating that all of its transitions are uncontrollable. As a consequence, the unique run of the 2CM reaches a halting configuration if and only if the unique run in the LHG, resulting from the vacuous strategy, reaches the target T. ∎

As a consequence of the previous theorem, the reachability control problem is also undecidable for the larger class of non-Zeno LHGs.

3. THE GLOBAL SEMI-ALGORITHM

The following theorem states the general procedure for solving the reachability control problem, based on the *controllable predecessor operator for reachability* $CPre^{\mathrm{R}}(\cdot)$, defined as follows.

For a set of states A, the operator $CPre^{\mathrm{R}}(A)$ returns the set of states from which the controller can ensure that the system reaches A within the next joint step. This can happen if there exists a strategy the controller can follow, all of whose consistent runs satisfy the following two properties: *(1)* if the run ever takes a discrete transition, either the first one leads to A or the run passes through A before taking it; and *(2)* if the run follows an admissible trajectory forever, then it must eventually pass through A. We then have:

$$\begin{aligned} CPre^{\mathrm{R}}(A) = \Big\{ s = \langle l, u\rangle \in InvS \,\Big|\, & \exists \sigma \forall r \in Runs(s,\sigma) : \\ & \text{if } r = s \xrightarrow{f,\delta,e} s' \cdots \text{ then } s' \in A \text{ or } \exists \delta' \in [0,\delta] \,.\, \langle l, f(\delta')\rangle \in A \\ & \qquad \text{and if } r = s \xrightarrow{f,\infty} \text{ then } \exists \delta' \geq 0 \,.\, \langle l, f(\delta')\rangle \in A \Big\}. \end{aligned}$$

In discrete games, the CPre operator used for solving reachability games is the same as the one used for the safety goal [Maler 2002]. In both cases, when the operator is applied to a set of states T, it returns the set of states from which Player 1 can force the game into T in one step. In hybrid games, the situation is different: a joint

[3]We are referring to the first scenario in the proof of Theorem 4.1, addressing the case $k_1 > k_2 > 0$.

step represents a complex behavior, extending over a (possibly) non-zero time interval. While the CPre for reachability only requires T to be visited once during such interval, CPre for safety requires that the entire behavior constantly remains in T. Hence, in Section 3.1 we present a novel algorithm for computing $\mathit{CPre}^{\mathrm{R}}$.

The following theorem states that the least fixpoint of the operator $\tau(X) \triangleq T \cup \mathit{CPre}^{\mathrm{R}}(X)$ provides a solution of the reachability control problem. Intuitively, each application of the τ extends (backward), by adding a single joint step, all the runs compatible with some winning strategy for the controller, until a fixpoint is reached.

THEOREM 3.1. *Let T be a polyhedron and $W^* = \mu W \,.\, T \cup \mathit{CPre}^{\mathrm{R}}(W)$, where μ denotes the least fixpoint. If the fixpoint is obtained in a finite number of iterations then the answer to the reachability control problem for target set T is positive if and only if $\mathit{InitS} \subseteq W^*$.*

Since $\mathit{CPre}^{\mathrm{R}}(\cdot)$ is effectively computable on polyhedra (as we show in the following), the results of Henzinger et al. [1995] imply that the above fixpoint may not be reached within a finite number of iterations. However, experiments such as the ones we describe in Section 7 suggest that it may converge in cases of practical relevance.

Before proving Theorem 3.1, let us recall the following lemma, which is an adaptation of Lemma 4.1 in [Alur et al. 1996], and states that any point reached by an admissible trajectory can be reached with a straight-line admissible trajectory as well. Notice that the original version of the lemma applies to differentiable trajectories, whereas our trajectories may not be differentiable in a finite set of time instants.

LEMMA 3.2. *For all points $p \in \mathit{Inv}(l)$, trajectories $f \in \mathit{Adm}(\langle l, p\rangle)$ and $\delta > 0$, let $c = \frac{f(\delta)-p}{\delta}$. Then, $c \in \mathit{Flow}(l)$.*

PROOF. We proceed by induction on the number of delays $\gamma \in (0, \delta)$ such that f is not differentiable in γ. If such number is zero, then the thesis follows immediately from Lemma 4.1 in [Alur et al. 1996]. Otherwise, by applying the inductive hypothesis to the two intervals $[0, \gamma]$, $[\gamma, \delta]$, we obtain that $f(\gamma) = p+\gamma c_1$ and $f(\delta) = p+\gamma c_1+(\delta-\gamma)c_2$, where $c_1 = \frac{f(\gamma)-p}{\gamma} \in \mathit{Flow}(l)$ and $c_2 = \frac{f(\delta)-f(\gamma)}{\delta-\gamma} \in \mathit{Flow}(l)$. In other words, $f(\delta) = p + \delta c$, where $c = \frac{\gamma}{\delta}c_1 + \frac{\delta-\gamma}{\delta}c_2$. Since c is a convex combination of c_1 and c_2, we obtain that $c \in \mathit{Flow}(l)$, hence the thesis. ∎

Proof of Theorem 3.1: *if*. Assume that $\mathit{InitS} \subseteq W^*$, we shall build a strategy that is winning from all initial states. Let $W_0 = T$ and, for all $n \geq 0$:

$$W_{n+1} = W_n \cup \mathit{CPre}^{\mathrm{R}}(W_n).$$

For $e = (l, J, l') \in \mathit{Edg}$ and $A, B \subseteq \mathit{InvS}$, define $\mathit{Jump}(A, e, B)$ as the set of valuations $w \in J$ such that $\langle l, w\lfloor_X\rangle \in A$ and $\langle l', w\lfloor_{X'}\rangle \in B$. For all $n \geq 0$, let σ_n be the strategy defined as follows, for all controllable transitions $e \in \mathit{Edg}_c$. Let $\sigma_0(e) = \emptyset$, and

$$\sigma_{n+1}(e) = \sigma_n(e) \cup \mathit{Jump}(W_{n+1} \setminus W_n, e, W_n).$$

We prove that, for all n, σ_n is a winning strategy from each state $s \in W_n$. We shall need the following lemma, whose proof is reported in the Appendix.

LEMMA 3.3. *For all $n \geq 0$ and states $s \in W_n$, all runs starting from s and consistent with σ_n reach T.*

Using Lemma 3.3 it is now immediate to prove the *if* direction of Theorem 3.1. Indeed, let $\sigma^* = \sigma_n$, where n is such that $W_n = W^*$. Then, Lemma 3.3 ensures that for any state $s \in W^*$ and any run $r \in \mathit{Runs}(s, \sigma^*)$, r eventually reaches a state in T. ∎

Proof of Theorem 3.1: *only if*. To prove the other direction, assume $s_0 \notin W^*$ and let $\overline{\sigma}$ be any strategy. We shall prove that there is a run $\overline{r}$ starting from s_0 and consistent with $\overline{\sigma}$ such that $States(\overline{r}) \cap T = \emptyset$. By definition of W^*, $s_0 \notin W^*$ implies that $s \notin T \cup CPre^{\mathrm{R}}(W^*)$. Therefore for all strategies, there exists a run from s_0 consistent with the strategy whose first joint step is completely contained in $\overline{W^*}$. Hence, there exists a run $r_0 \in Runs(s_0, \overline{\sigma})$ such that either $r_0 = s_0 \xrightarrow{f_0,\infty}$ and $f_0(\delta) \in \overline{W^*}$ for all $\delta \geq 0$, or $r_0 = s_0 \xrightarrow{f_0,\delta_0,e_0} s_1 \cdots$, $\langle loc(s_0), f_0(\delta') \rangle \in \overline{W^*}$ for all $\delta' \in [0, \delta_0]$, and $s_1 \in \overline{W^*}$. In the first case, we set $\overline{r} = r_0$ and we are done, since $\overline{W^*} \cap T = \emptyset$. In the second case, since $s_1 \notin W^*$, we can repeat the same reasoning starting from s_1, and obtain a new run $r_1 \in Runs(s_1, \overline{\sigma})$ with the same properties as r_0. Once again, if $r_1 = s_1 \xrightarrow{f_1,\infty}$ and $f_1(\delta) \in \overline{W^*}$ for all $\delta \geq 0$, we obtain the desired run $\overline{r}$ by concatenating the first joint step of run r_0 with r_1, i.e., $\overline{r} = s_0 \xrightarrow{f_0,\delta,e} s_1 \xrightarrow{f_1,\infty}$. Otherwise, $r_1 = s_1 \xrightarrow{f_1,\delta_1,e_1} s_2 \cdots$ and $f_1(\delta') \in \overline{W^*}$, for all $\delta' \in [0, \delta_1]$ and $s_2 \in \overline{W^*}$. The concatenation of the first joint step of r_0 with r_1 is a run $s_0 \xrightarrow{f_0,\delta_0,e_0} s' \xrightarrow{f_1,\delta_1,e_1} s_2 \cdots$, which is consistent with $\overline{\sigma}$ and does not lead to T within the first two joint steps. By iterating the above reasoning and concatenating the first joint steps of the runs r_i, with $i \geq 0$, we can form a run $\overline{r} \in Runs(s_0, \overline{\sigma})$ which is composed either of a finite number of joint steps ending with an infinite time step (if there is an i such that $r_i = s_i \xrightarrow{f_i,\infty}$), or of an infinite number of joint steps. In either case, each joint step of the resulting run is completely contained in $\overline{W^*}$, which in turn is disjoint from T, hence the conclusion follows. ∎

3.1. Computing the Predecessor Operator

In order to compute the predecessor operator, we introduce the *Must Reach While Avoiding* operator, denoted by RWA^{M}. Given a location l and two sets of variable valuations U and V, $RWA^{\mathrm{M}}_l(U, V)$ contains the set of valuations from which all continuous trajectories of the system reach U while avoiding V [4]. Formally, we have:

$$RWA^{\mathrm{M}}_l(U,V) = \Big\{ u \in Val(X) \,\Big|\, \forall f \in Adm(\langle l, u\rangle)\, \exists \delta \geq 0 : f(\delta) \in U \text{ and } \forall\, 0 \leq \delta' \leq \delta : f(\delta') \notin V \Big\}. \quad (2)$$

The definition requires trajectories to avoid V even in the time instant when U is reached, i.e., reaching a point in $U \cap V$ is not acceptable. Hence, it holds $RWA^{\mathrm{M}}_l(U, V) = RWA^{\mathrm{M}}_l(U \setminus V, V)$ and in the following we can assume w.l.o.g. that U and V are disjoint.

The operator $CPre^{\mathrm{R}}(\cdot)$ can now be reformulated, by means of the operator $RWA^{\mathrm{M}}_l(\cdot, \cdot)$, based solely on the geometric properties of the admissible trajectories. Let $B_l = Pre_{\mathrm{u}}(\overline{A})|_l$ be the set of states of location l, where the environment can take a discrete transition leading outside A and, similarly, $C_l = Pre_{\mathrm{c}}(A)|_l$ be the set of states of l, where the controller can take a discrete transition leading to A. According to the defition, a state s of location l belongs to $CPre(A)$ if the controller can force the system into A within one joint step, no matter what the environment does. This occurs if, for every possible trajectory chosen by the environment, one of the following conditions holds: (i) the system reaches $A|_l$ while avoiding $B_l \setminus A|_l$, thus without giving the environment any chance to take an action leading outside A; (ii) the system reaches a point in $C_l \setminus B_l$, from where the controller can force the system into A, while avoiding $B_l \setminus A|_l$; or (iii) the trajectory exits from the invariant $Inv(l)$ meanwhile avoiding $B_l \setminus A|_l$, but no

[4] In the temporal logic CTL, we have $RWA^{\mathrm{M}}(U, V) \equiv \forall \overline{V}\ \mathcal{U}\ (U \wedge \overline{V})$.

point in $A\lfloor_l \cup (C_l \setminus B_l)$ is ever reached. In this last case, the well-formedness condition ensures that, before the trajectory reaches $\overline{Inv(l)}$, the environment must take some discrete transition, which can only lead to A.

The following lemma formalizes the above intuition. We say that a set of states $A \subseteq S$ is *polyhedral* if for all $l \in Loc$, the projection $A\lfloor_l$ is a polyhedron.

LEMMA 3.4. *For all polyhedral sets of states $A \subseteq InvS$, let $B_l = Pre_{\rm u}(\overline{A})\lfloor_l$ and $C_l = Pre_{\rm c}(A)\lfloor_l$. We, then, have:*

$$CPre^{\rm R}(A) = InvS \cap \bigcup_{l \in Loc} \{l\} \times RWA^{\rm M}_l\big(A\lfloor_l \cup C_l \setminus B_l \cup \overline{Inv(l)}, B_l \setminus A\lfloor_l\big).$$

PROOF. [$\subseteq$] Let $s = \langle l, u\rangle$ and assume that $u \notin RWA^{\rm M}_l\big(A\lfloor_l \cup C_l \setminus B_l \cup \overline{Inv(l)}, B_l \setminus A\lfloor_l\big)$, then there exists a trajectory $f \in Adm(s)$ such that for all $\delta \geq 0$, either $f(\delta) \notin A\lfloor_l \cup (C_l \setminus B_l) \cup \overline{Inv(l)}$ or there exists $\delta' \in [0, \delta]$ with $f(\delta') \in B_l \setminus A\lfloor_l$. We prove that $s \notin CPre^{\rm R}(A)$.

Let us first consider the case where $f(\delta) \notin A\lfloor_l \cup (C_l \setminus B_l) \cup \overline{Inv(l)}$ for all $\delta \geq 0$. In this case, $span(f, l) = \infty$, since f never exits from $Inv(l)$. Let σ be any strategy, if σ never prescribes any controllable transition along the trajectory f, then the run $s \xrightarrow{f,\infty}$, which follows f forever, is consistent with σ. Our assumption ensures that this run never reaches $A\lfloor_l$, and hence $s \notin CPre^{\rm R}(A)$.

If instead σ forces some discrete transition $e \in Edg_{\rm c}$ to be taken along f, a run of the type $s \xrightarrow{f,\overline{\delta}} \overline{s} \xrightarrow{e} s' \cdots$ is consistent with σ. Since, by assumption, $f(\overline{\delta}) \notin C_l \setminus B_l$, there are two cases: either $f(\overline{\delta}) \notin C_l$, in which case $s' \notin A$, or $f(\overline{\delta}) \in C_l \cap B_l$. In the latter case, however, the run $s \xrightarrow{f,\overline{\delta}} \overline{s} \xrightarrow{e'} s'' \cdots$, with $e' \in Edg_{\rm u}$, is also consistent with σ and $s'' \notin A$ since $f(\overline{\delta}) \in B_l$. In either case it follows that $s \notin CPre^{\rm R}(A)$.

Let us now consider the case where there is a $\delta \geq 0$ with $f(\delta) \in A\lfloor_l \cup (C_l \setminus B_l) \cup \overline{Inv(l)}$ and for some $\delta' \in [0, \delta]$, $f(\delta') \in B_l \setminus A\lfloor_l$. Observe that we can assume $\delta > 0$. Otherwise either u would trivially belong to $RWA^{\rm M}_l\big(A\lfloor_l \cup C_l \setminus B_l \cup \overline{Inv(l)}, B_l \setminus A\lfloor_l\big)$ or it would hold $s \notin InvS$, in which case $s \notin CPre^{\rm R}(A)$ by definition. Let $\Delta_f = \{\delta \geq 0 \mid f(\delta) \in A\lfloor_l \cup (C_l \setminus B_l) \cup \overline{Inv(l)}\}$ and $\hat{\delta} = \inf \Delta_f$.

If $\hat{\delta} \in \Delta_f$, then $\hat{\delta} > 0$ by the observation above. Moreover, for all $\delta \in [0, \hat{\delta})$, $f(\delta) \notin A\lfloor_l \cup (C_l \setminus B_l) \cup \overline{Inv(l)}$ and $f(\delta') \in B_l \setminus A\lfloor_l$ for some $\delta' \in [0, \hat{\delta})$. Therefore, given an arbitrary strategy σ, if σ does not prescribe any controllable transition along f in the interval $[0, \hat{\delta})$, then there is a run of the form $s \xrightarrow{f,\delta'} s' \xrightarrow{e} s'' \cdots$ with $e \in Edg_{\rm u}$ and consistent with σ. Since $f(\delta') \in B_l \setminus A\lfloor_l$, $s'' \notin A$ as desired. If, on the other hand, σ forces a controllable transition along f in the interval $[0, \hat{\delta})$, then it is either taken from a point belonging to $\overline{C_l}$, hence leading to $\overline{A}$, or it must be taken from a point v belonging to $C_l \cap B_l$ and the environment can always take an uncontrollable transition from state $v \in B_l$ which leads to $\overline{A}$.

Finally, consider the case where $\hat{\delta} \notin \Delta_f$. Then for all $\delta \in [0, \hat{\delta}]$, $f(\delta) \notin A\lfloor_l \cup (C_l \setminus B_l) \cup \overline{Inv(l)}$ and $f(\delta') \in B_l \setminus A\lfloor_l$ for some $\delta' \in [0, \hat{\delta}]$. By a reasoning similar to the previous case, for any strategy σ we can build a run from s consistent with σ which takes an uncontrollable transition leading to $\overline{A}$ in the interval $[0, \hat{\delta}]$. Again, we can conclude that $s \notin CPre^{\rm R}(A)$.

[$\supseteq$] Assume that $u \in RWA^{\rm M}_l\big(A\lfloor_l \cup C_l \setminus B_l \cup \overline{Inv(l)}, B_l \setminus A\lfloor_l\big)$ and let $s = \langle l, u\rangle$. Define σ so that, for every $e \in Edg_{\rm c}$, $\sigma(e) = Jump(\{l\} \times C_l \setminus \{l\} \times B_l, e, A)$. We shall show that every run consistent with σ leads to A within one joint step. Let $r \in Runs(s, \sigma)$ and let

f be the trajectory followed by r before the first discrete step, if r ever takes a discrete step. There are two cases: either (i) $r = s \xrightarrow{f,\infty}$ or (ii) $r = s \xrightarrow{f,\delta,e} s' \cdots$. In case (i), f eventually reaches A as desired. Indeed, f cannot reach $\overline{Inv(l)}$, since r is a single infinite time step along f which must then satisfy $span(f, loc(s)) = \infty$. Moreover, by consistency of r w.r.t. σ, f cannot reach $C_l \setminus B_l$ either, otherwise σ would eventually force a discrete transition.

Then, consider case (ii). If $s' \in A$ then we are done. Assume that $s' \notin A$, then, by consistency of r w.r.t. σ, $e \in Edg_{\mathrm{u}}$ and $f(\delta) \in B_l$. Consistency w.r.t. σ also ensures that for all $\delta' \in [0, \delta]$, $f(\delta') \notin C_l \setminus B_l$, otherwise a controllable transition would necessarily have been taken before reaching $\langle loc(s), f(\delta)\rangle$. Moreover, since $\delta \in span(f, loc(s))$, for all $\delta' \in [0, \delta]$, $f(\delta') \in Inv(l)$. Therefore, since $u \in RWA^{\mathrm{M}}_l\big(A\lfloor_l \cup C_l \setminus B_l \cup \overline{Inv(l)}, B_l \setminus A\lfloor_l\big)$, there must be a $\delta' \in [0, \delta]$ with $f(\delta') \in A\lfloor_l$, hence the conclusion. ∎

4. THE LOCAL ALGORITHM

The previous section reduces the reachability control problem to the computation of the operator RWA^{M}. Let us start by examining the basic properties of RWA^{M}.

Example 4.1. As witnessed by Figure 4(a), the first argument of RWA^{M} does not distribute over union; in other words $RWA^{\mathrm{M}}_l(U_1 \cup U_2, V) \neq RWA^{\mathrm{M}}_l(U_1, V) \cup RWA^{\mathrm{M}}_l(U_2, V)$. In particular, for the polyhedra in Figure 4(a), with the flow constraint $F = Flow(l)$ shown in the left-hand side box, we have that the area called R_3 belongs to $RWA^{\mathrm{M}}_l(U_1 \cup U_2, V)$ but it does not belong to either $RWA^{\mathrm{M}}_l(U_1, V)$ or $RWA^{\mathrm{M}}_l(U_2, V)$. Hence, computing $RWA^{\mathrm{M}}_l(U, V)$ for convex U (a relatively simple task) does not extend easily to general polyhedra.

On the other hand, the following proposition allows us to restrict the second argument of RWA^{M} to being a *convex* polyhedron.

PROPOSITION 4.2. *For all polyhedra U, V_1, and V_2 it holds that*

$$RWA^{\mathrm{M}}_l(U, V_1 \cup V_2) = RWA^{\mathrm{M}}_l(U, V_1) \cap RWA^{\mathrm{M}}_l(U, V_2).$$

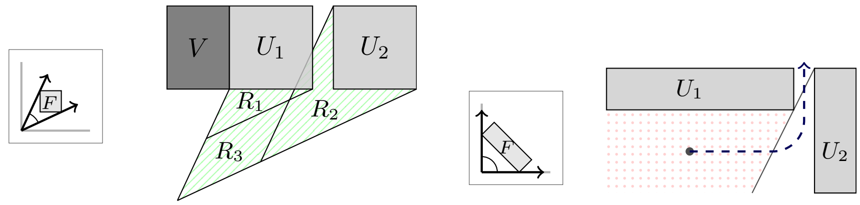
Indeed, all trajectories from a given point avoid the union of two polyhedra if and only if those same trajectories avoid each of them.

Example 4.3. It is easy to see that it is not possible to restrict the analysis from arbitrary trajectories to straight-line trajectories. In Figure 4(b), the dotted area contains the set of points that must reach $U_1 \cup U_2$ following straight-line trajectories. On the other hand, $RWA^{\mathrm{M}}_l(U_1 \cup U_2, \emptyset) = U_1 \cup U_2$, because all other points (including those in the dotted area) can avoid $U_1 \cup U_2$ by passing through the gap between U_1 and U_2.

Here, we show how to compute RWA^{M} based on the operator which is used to solve *safety* control problems: the *May Reach While Avoiding* operator $RWA^{\mathrm{m}}_l(U, V)$, returning the set of states from which *there exists* a trajectory that reaches U while avoiding V. Formally:

$$RWA^{\mathrm{m}}_l(U, V) = \Big\{u \in Val(X) \;\Big|\; \exists f \in Adm(\langle l, u\rangle),\ \delta \geq 0 : \\ f(\delta) \in U \text{ and } \forall\, 0 \leq \delta' < \delta : f(\delta') \in \overline{V} \cup U\Big\}.$$

In safety control problems, RWA^{m} is used to compute the states from which the environment may reach an unsafe state (in U) while avoiding the states from which the controller can take a transition to a safe state (in V). Notice that RWA^{m} is a classical operator, known under different names such as *Reach* [Tomlin et al. 2000], *Un-*



(a) $RWA_l^{\mathrm{M}}(U_1 \cup U_2, V) \neq RWA_l^{\mathrm{M}}(U_1, V) \cup RWA_l^{\mathrm{M}}(U_2, V)$.

(b) Straight-line trajectories are not sufficient to avoid $U_1 \cup U_2$.

Fig. 4. Basic properties of RWA^{M}. The boxes on the left represent the convex polyhedron $F = Flow(l)$ in the $(\dot{x}, \dot{y})$ plane. Thick arrows represent the *extremal directions* of flow.

avoid_Pre [Balluchi et al. 2003], and *flow_avoid* [Wong-Toi 1997]. We recently gave the first sound and complete algorithm for computing it on LHGs [Benerecetti et al. 2013].

In the rest of this section, we consider a fixed location $l \in Loc$ and we omit the l subscript whenever possible. For a polyhedron G and $p \in G$, we say that p is *t-bounded in* G if all admissible trajectories starting from p eventually exit from G. Formally, p is t-bounded if for all $f \in Adm(\langle l, p\rangle)$ there exists $\delta \geq 0$ such that $f(\delta) \notin G$. We denote by $t\text{-}bnd(G)$ the set of points of G that are t-bounded in it, and we say that G is *t-bounded* if all points $p \in G$ are t-bounded in G.

Example 4.4. Consider the L-shaped polyhedron G depicted in Figure 5, where the only flow direction is upwards. Point p_1 is not t-bounded in G, because G extends indefinitely upwards from p_1. Point p_2 is t-bounded because it sits on the upper boundary of G, and finally p_3 is t-bounded in G, as the trajectory that starts from p_3 eventually (but not immediately) exits from G. The gray region of G is $t\text{-}bnd(G)$.

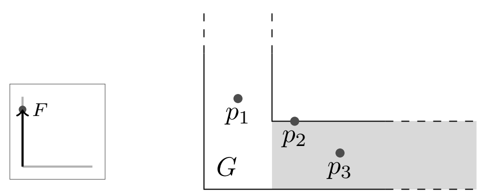


Fig. 5. A non-convex polyhedron containing t-bounded and non-t-bounded points.

We now show how to relate RWA^{M} and RWA^{m}, by exploiting the following idea. First, notice that all points in U belong to $RWA^{\mathrm{M}}(U, V)$ by definition. Now, the content of $RWA^{\mathrm{M}}(U, V)$ can be partitioned into two regions: the first region is U; the second region must be t-bounded, because each point in the second region must eventually reach U. If we can find a polyhedron $Over$ that over-approximates $RWA^{\mathrm{M}}(U, V)$ and such that $Over \setminus U$ is t-bounded, we can use RWA^{m} to refine it. Precisely, we can use RWA^{m} to identify and remove the points of $Over$ that may leave $Over$ without hitting U first.

If $Over \setminus U$ is not t-bounded, the above technique does not work, because RWA^{m} cannot identify (and remove) the points that may remain forever in $Over$ without ever reaching U. This idea is formalized by the following result.

THEOREM 4.5. *For all disjoint polyhedra U and V, such that V is convex, let $Over$ be a polyhedron such that:* (i) $RWA^{\mathrm{M}}(U, V) \subseteq Over \subseteq \overline{V}$ *and* (ii) $Over \setminus U$ *is t-bounded.*

Then,

$$RWA^{\mathrm{M}}(U,V) = Over \setminus RWA^{\mathrm{m}}(\overline{Over}, U). \tag{3}$$

PROOF. [$\subseteq$] Let $u \in RWA^{\mathrm{M}}(U,V)$. By assumption *(i)*, it holds $u \in Over$. We prove that $u \notin RWA^{\mathrm{m}}(\overline{Over}, U)$. Assume the contrary; according to the definition of RWA^{m}, there exist a trajectory $f \in Adm(\langle l, u\rangle)$ and a delay $\delta \geq 0$ such that $f(\delta) \in \overline{Over}$ and $f(\delta') \in \overline{U} \cup \overline{Over}$ for all $0 \leq \delta' < \delta$. Again by assumption *(i)*, it holds $U \subseteq Over$ and hence $f(\delta') \in \overline{U}$ for all $0 \leq \delta' \leq \delta$. Since $\overline{Over} \subseteq \overline{RWA^{\mathrm{M}}(U,V)}$, the trajectory f leads from u to a point outside $RWA^{\mathrm{M}}(U,V)$, without passing through U.

Let f' be a trajectory witnessing the fact that $f(\delta) \notin RWA^{\mathrm{M}}(U,V)$. The trajectory obtained by concatenating f at time δ and f' is a witness for $u \notin RWA^{\mathrm{M}}(U,V)$, which is a contradiction.

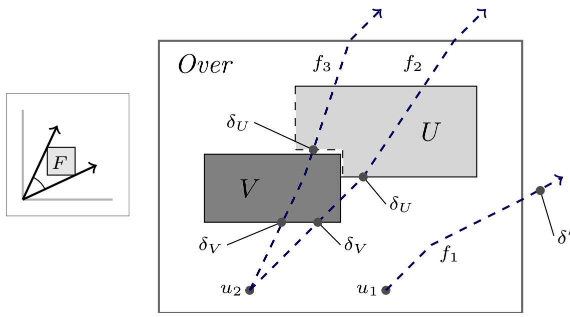


Fig. 6. Different ways of *not* belonging to $RWA^{\mathrm{M}}(U,V)$.

[$\supseteq$] Let $u \notin RWA^{\mathrm{M}}(U,V)$. It is immediate that $u \notin U$. We prove that $u \notin Over \setminus RWA^{\mathrm{m}}(\overline{Over}, U)$. If $u \notin Over$, we are done. Hence, assume that $u \in Over$. Since $u \notin RWA^{\mathrm{M}}(U,V)$, by definition there is a trajectory $f \in Adm(\langle l, u\rangle)$ such that for all $\delta \geq 0$, if $f(\delta) \in U$ the there is a previous time $\delta' \leq \delta$ such that $f(\delta') \in V$. We distinguish two cases:

— First, assume that the trajectory f never reaches U (see trajectory f_1 in Figure 6). By assumption *(ii)*, there exists $\delta' \geq 0$ such that $f(\delta') \notin Over \setminus U$. Since $f(\delta') \notin U$, we conclude $f(\delta') \notin Over$. As a consequence, it holds $u \in RWA^{\mathrm{m}}(\overline{Over}, U)$, and we are done.

— Otherwise, let $D_U = \{\delta \geq 0 \mid f(\delta) \in U\} \neq \emptyset$ and $\delta_U = \inf D_U$. There are two cases: first assume $\delta_U \in D_U$ (as in trajectory f_2 in Figure 6). Since $f(\delta_U) \in U$, there exists a previous time $\delta' \leq \delta_U$ with $f(\delta') \in V$. This implies that f reaches V (and hence $\overline{Over}$) at time δ' while remaining outside U up until δ' (included). As a consequence, $u \in RWA^{\mathrm{m}}(\overline{Over}, U)$ and we are done.
Next, assume $\delta_U \notin D_U$ (as in trajectory f_3 in Figure 6). Let $D_V = \{\delta \geq 0 \mid f(\delta) \in V\}$. Since D_U is not empty, neither is D_V. Let $\delta_V = \inf D_V$. If $\delta_V < \delta_U$, there exists a time between δ_V and δ_U when f reaches V (and hence $\overline{Over}$). Since f remains in $\overline{U}$ until δ_U, we can conclude that $u \in RWA^{\mathrm{m}}(\overline{Over}, U)$.
Otherwise, $\delta_V \geq \delta_U$. However, assuming δ_V strictly larger than δ_U leads to an immediate contradiction, so in fact $\delta_V = \delta_U$. Now, if $\delta_V \in D_V$, then it immediately follows that $u \in RWA^{\mathrm{m}}(\overline{Over}, U)$. Otherwise, there are elements of D_V arbitrarily close to δ_V. Hence, $f(\delta_V) \in cl(V)$. Let $\hat{\delta} \in D_V$, define a trajectory f' as follows: f'

coincides with f up to time δ_V; then, f' proceeds along a straight line from $f(\delta_V)$ to $f(\hat{\delta})$; finally, it continues as f after time $\hat{\delta}$. By Lemma 3.2, $f' \in Adm(\langle l, u \rangle)$. Since $f(\delta_V) \in cl(V)$ and $f(\hat{\delta}) \in V$, by the convexity of V it holds that $f'(\delta) \in V$ for all $\delta_V < \delta \leq \hat{\delta}$. Therefore, at all times up to $\hat{\delta}$ (included), f' remains in $\overline{U} \cup \overline{Over}$. Once again we obtain that $u \in RWA^{\mathrm{m}}(\overline{Over}, U)$. ∎

Example 4.6. An example of the application of Theorem 4.5 is depicted in Figure 7, where U and V are the gray boxes and $Over$ is the outer box, excluding V. The set $RWA^{\mathrm{m}}(\overline{Over}, U)$ can be divided in two areas: area X_1 contains the points that may reach V (which is a part of $\overline{Over}$) while avoiding U, and area X_2 contains the points that may exit $Over$ through its top and right sides. Following Equation 3, we remove X_1 and X_2 from $Over$, and we are left with the region $U \setminus V$ and the two regions R_1 and R_2, whose points are forced to enter U while avoiding V, as requested by $RWA^{\mathrm{M}}(U, V)$.

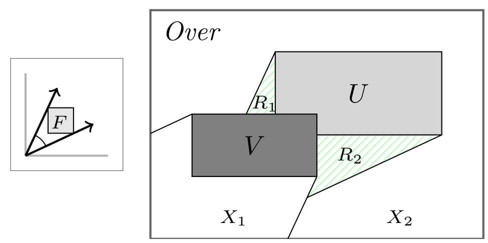


Fig. 7. Relationship between RWA^{M} and RWA^{m}.

The results above ensure that, if we can effectively compute the operator RWA^{m} on polyhedra and we start from a suitable over-approximation for $RWA^{\mathrm{M}}(U, V)$, then we can also effectively compute $RWA^{\mathrm{M}}(U, V)$, by applying Equation 3. The operator RWA^{m} is shown to be computable by Benerecetti et al. [2013], whereas Sections 5 and 6 describe techniques for computing the desired over-approximation. By Lemma 3.4 this, in turn, allows us to compute $CPre^{\mathrm{R}}$, leading to the following theorem.

THEOREM 4.7. *For all polyhedral sets of states A, $CPre^{\mathrm{R}}(A)$ is computable.*

Notice that the above result provides no guarantee of termination for the global fixpoint in Theorem 3.1. In particular, it does not imply semi-decidability of the reachability control problem, as the fixpoint may not be reached within ω iterations of $CPre^{\mathrm{R}}$.

5. ON BOUNDED POLYHEDRA

In order to apply Theorem 4.5, we must be able to compute a suitable polyhedron t-bounded w.r.t. some convex (and bounded) polyhedron F. Since, however, boundedness w.r.t. arbitrary trajectories is hard to directly reason about, we shall relate it to *geometric* boundedness, i.e., boundedness w.r.t. straight-line trajectories. The objective of this section is, therefore, to provide properties connecting these two notions.

For a (possibly non-convex) polyhedron G and a convex polyhedron F, we say that G is *bounded w.r.t. F* if for all $p \in G$ and all $c \in F$ there exists a constant $\delta \geq 0$ such that $p + \delta c \notin G$. Intuitively, G is bounded w.r.t. F if all straight lines starting from G and whose slope belongs to F eventually exit from G. Clearly, if the origin is contained in F, then F admits stationary trajectories and the following observation follows.

PROPOSITION 5.1. *If F is a convex polyhedron containing the origin, then no polyhedron is bounded w.r.t. F.*

The following necessary condition for t-boundedness is immediate, since straight lines are a special case of trajectories.

PROPOSITION 5.2. *If a polyhedron is t-bounded w.r.t F, then it is bounded w.r.t. F.*

We now turn our attention to sufficient conditions for t-boundedness w.r.t. a closed and convex polyhedron F. We can prove that, when an admissible trajectory lies in P in an infinite sequence of diverging time instants, there is a straight trajectory that always lies in P.

Given two convex polyhedra P and F, and a point x, for all $c \in F$, define $reach_x^P(c)$ as the infimum of the delays $\delta \geq 0$ such that $x + \delta c \in P$, or ∞ if no such δ exists. Intuitively, $reach_x^P(c)$ is the minimum time needed to reach P from x along direction c. Clearly, if P is a hyperplane of equation $ax = b$, then $reach_x^P(c) = \frac{b-ax}{ac}$.

LEMMA 5.3. *Let F be a closed and bounded convex polyhedron, P a convex polyhedron, $x \in P$, and $f \in Adm(x, F)$. In addition, let $\{\delta_i\}_{i\in\mathbb{N}}$ be a diverging sequence of time instants. If $f(\delta_i) \in P$ for all $i \in \mathbb{N}$ then there exists $c \in F$ such that $x + \delta c \in P$ for all $\delta \geq 0$.*

PROOF. For all $\delta \geq 0$, let $g(\delta) = \frac{f(\delta)-x}{\delta}$. By Lemma 3.2, it holds that $g(\delta) \in F$. The infinite sequence $\{g(\delta_i)\}_{i\in\mathbb{N}}$ takes value in the compact set F. Let $c_i = g(\delta_i)$, by Bolzano-Weierstrass there exists a subsequence $\{c_{i_j}\}_{j\in\mathbb{N}}$ that converges to a point $\hat{c} \in F$. Let $exit_x(c) = \sup\{\delta \geq 0 \mid x + \delta c \in P\}$ be the time needed to exit from P following the direction c. To obtain our thesis it suffices to show that $exit_x(\hat{c}) = \infty$. Notice that for all $i \geq 0$ it holds that $exit_x(c_i) > \delta_i$. Let $P_1, \ldots, P_m$ be the supporting hyperplanes of P, where P_h is defined by $a_h x = b_h$ for all $h = 1, \ldots, m$. When $exit_x(c) < \infty$ we have that: *(i)* $exit_x(c) = \min\{reach_x^{P_h}(c) \mid h = 1, \ldots, m$ and $reach_x^{P_h}(c) > 0\}$, where $\min \emptyset = 0$; *(ii)* since $reach_x^{P_h}(c)$ is continuous in c, so is $exit_x(c)$. Notice that if $reach_x(c_i) = \infty$ for some i, then the thesis follows immediately. Otherwise, continuity of $exit_x$ implies the following:

$$exit_x(\hat{c}) = exit_x(\lim_j c_{i_j}) = \lim_j exit_x(c_{i_j}) \geq \lim_j \delta_{i_j} = \infty. \qquad \blacksquare$$

Lemma 5.3 allows us to state a sufficient condition for being t-bounded w.r.t. F.

LEMMA 5.4. *If a convex polyhedron is bounded w.r.t. a closed convex F then it is t-bounded w.r.t. F.*

PROOF. Assume P is a convex polyhedron not t-bounded w.r.t F. Then there exist a point $p \in P$ and an admissible trajectory $f \in Adm(p, F)$, such that $f(\delta) \in P$, for all $\delta \geq 0$. Clearly, $f \in Adm(p, F)$ as well. Then, for any strictly increasing diverging sequence of (non negative) time instants $\{\delta_i\}_{i\in\mathbb{N}}$, it holds $f(\delta_i) \in P$. Lemma 5.3 applied to F, P, p, f and $\{\delta_i\}_{i\in\mathbb{N}}$ gives us a $c \in F$ such that $p + \delta c \in P$, for all $\delta \geq 0$. As a consequence, P is not bounded w.r.t. F, hence the thesis. $\blacksquare$

Note that, when F is not closed, being bounded w.r.t. F is no longer sufficient for a polyhedron to be t-bounded w.r.t F, as shown by the following example.

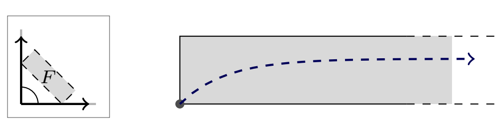


Fig. 8. On the right, a polyhedron which is bounded w.r.t. F but not t-bounded w.r.t F, and a trajectory that remains forever in it (see Example 5.5).

Example 5.5. Consider the unbounded polyhedron P shown on the r.h.s. of Figure 8. The dashed contour of F (on the l.h.s. of the figure) indicates that F is topologically open, so that its extremal directions $(1, 0)$ and $(0, 1)$ are not proper (i.e., they do not belong to F). It turns out that P is bounded w.r.t. F, because all straight lines whose slope belongs to F eventually exit from it, but it is not t-bounded w.r.t. F. The figure shows a trajectory that remains forever in P. Its slope approaches asymptotically the extremal direction $(1, 0)$.

Lemma 5.4, together with Proposition 5.2, shows that t-boundedness is in fact equivalent to geometric boundedness, in case F is closed and convex.

COROLLARY 5.6. *A convex polyhedron is t-bounded w.r.t a closed convex F if and only if it is bounded w.r.t. F.*

We conclude the section with the following theorem, which lifts the necessary and sufficient condition for t-boundedness from convex polyhedra to general polyhedra.

THEOREM 5.7. *A polyhedron G is t-bounded w.r.t. a closed convex F if and only if each convex polyhedron $P \in [\![G]\!]$ is t-bounded w.r.t. F.*

PROOF. [*only if*] Clearly, if from every point in G each trajectory admissible w.r.t. F eventually leaves G, then each trajectory admissible w.r.t. F starting from a convex polyhedron $P \in [\![G]\!]$ also leaves P. Hence the thesis.
[*if*] If G is empty, the result is trivially true. Assuming $\mathbf{0} \in F$ leads to a contradiction. Indeed, if this were the case, by Proposition 5.1, no polyhedron could be bounded w.r.t. F and, by Lemma 5.4, each convex polyhedron in $P \in [\![G]\!]$ would not be t-bounded w.r.t. F, contradicting the hypothesis. Therefore, we can assume that $\mathbf{0} \notin F$.

We now proceed by induction on the cardinality of $[\![G]\!]$. If the cardinality is 1, the thesis immediately follows.

Let $|[\![G]\!]| > 1$ and pick an arbitrary $P \in [\![G]\!]$. By inductive hypothesis, $G \setminus P$ is t-bounded w.r.t. F. By contradiction, assume that G is not t-bounded w.r.t. F, and let $p \in G$ and $f \in Adm(p, F)$ be such that $f(\delta) \in G$ for all $\delta \geq 0$. If f eventually remains forever in $G \setminus P$ (i.e., there is $\delta \geq 0$ such that for all $\delta' \geq \delta$ it holds $f(\delta') \in G \setminus P$), we conclude that $G \setminus P$ is not t-bounded w.r.t. F, contradicting the inductive hypothesis. If f eventually remains forever in P, the contradiction follows form the assumption that P is t-bounded w.r.t. F. Therefore, f enters and exits from P infinitely often. Formally, for all $\delta \geq 0$ there exist $\delta', \delta'' \geq \delta$ such that $f(\delta') \in P$ and $f(\delta'') \in G \setminus P$. Since $[\![G \setminus P]\!]$ is a finite set of convex polyhedra, there must be a convex polyhedron $P' \in [\![G \setminus P]\!]$ which is adjacent to P and such that f crosses the boundary between P and P' infinitely often.

For any two polyhedra A and B, we define their *boundary* to be

$$bndry(A, B) = (cl(A) \cap B) \cup (A \cap cl(B)).$$

It is not hard to see that, if both A and B are convex polyhedra, then so is $bndry(A, B)$. Let, now, $b = bndry(P, P')$ be the boundary between P and P', such that f crosses b infinitely often, i.e., for all $\delta \geq 0$ there is $\delta' > \delta$ such that $f(\delta') \in b$. Since both P and P' are convex and bounded w.r.t. F by assumption, then b is both convex and t-bounded w.r.t. F. Let $\{\delta_i\}_{i \in \mathbb{N}}$ be a sequence of time instants such that *(i)* $f(\delta_i) \in b$ and *(ii)* $\delta_{i+1} \geq \delta_i + 1$.

By Corollary 5.6, b must be bounded w.r.t. F. Since, however, $\{\delta_i\}_{i \in \mathbb{N}}$ is an increasing diverging sequence and $f(\delta_i) \in b$, for all $i \in \mathbb{N}$, Lemma 5.3 gives us a straight direction c belonging to F with $f(\delta_0) + \delta\, c \in b$, for all $\delta \geq 0$. This contradicts the fact that b is bounded w.r.t. F. Hence, we conclude the thesis. ∎

5.1. Computing Boundedness For Convex Polyhedra

We conclude this section by providing effective ways to test for boundedness of a convex polyhedron P and to compute the set of points of a convex polyhedron which are not t-bounded.

We say that a vector r is a *ray* (a.k.a. *direction of unboundedness*) of a polyhedron G if there exists a point $x \in G$ such that for all $\delta \geq 0$ it holds $x + \delta r \in G$. We denote by $Rays(G)$ the set of rays of G.

Convex polyhedra admit two finite representations, in terms of *constraints* or *generators*. Libraries like PPL [Bagnara et al. 2008] maintain both representations for each convex polyhedron and efficient algorithms exist for keeping them synchronized [Chernikova 1968; Verge 1992]. The constraint representation refers to the set of linear inequalities whose solutions are the points of the polyhedron. The generator representation consists in three finite sets of *points*, *closure points*, and *rays*, that generate all points in the polyhedron by linear combination. More precisely, for each convex polyhedron $P \subseteq \mathbb{R}^n$ there exists a triple (V, C, R) such that V, C, and R are finite sets of points in $\mathbb{R}^n$, and $x \in P$ if and only if it can be written as

$$\sum_{v \in V} \alpha_v \cdot v + \sum_{c \in C} \beta_c \cdot c + \sum_{r \in R} \gamma_r \cdot r, \tag{4}$$

where all coefficients α_v, β_c and γ_r are non-negative reals, $\sum_{v \in V} \alpha_v + \sum_{c \in C} \beta_c = 1$, and there exists $v \in V$ such that $\alpha_v > 0$. We call the triple (V, C, R) a *generator system* for P.

Intuitively, the elements of V are the proper vertices of the polyhedron P, the elements of C are vertices of the topological closure of P that do not belong to P, and each element of R represents a direction of unboundedness of P. In the following, we tacitly assume that generator systems are minimal, in the sense that no element from V, C, or R can be removed without affecting the corresponding polyhedron. Moreover, we assume w.l.o.g. that the sets V, C, and R are mutually disjoint.[5]

For a convex polyhedron P, let O_P denote its *characteristic cone*, i.e., the closed polyhedron generated by the origin $\mathbf{0}$ and all the rays of P. If (V_P, C_P, R_P) is the generator system for P, then $(\{\mathbf{0}\}, \emptyset, R_P)$ is the generator system for O_P. The following theorem shows how we can effectively and efficiently test whether P is bounded w.r.t. F. For two sets of points A and B, the *Minkowski sum* $A \oplus B$ is $\{a + b \mid a \in A, b \in B\}$.

THEOREM 5.8. *For all convex polyhedra P and F, P is bounded w.r.t. F iff $O_P \cap F = \emptyset$.*

PROOF. [$\Rightarrow$] By hypothesis, for all $p \in P$ and for all $c \in F$ there exists $\delta \geq 0$ such that $p + \delta \cdot c \notin P$. By Proposition 5.1 we have that $\mathbf{0} \notin F$. Let $c \in F$, we show that $c \notin O_P$. Assume by contradiction that $c \in O_P$, we can write $c = 1 \cdot \mathbf{0} + \sum_{r \in R_p} \beta_r r = \sum_{r \in R_p} \beta_r r$. Now, let $x \in V_p$ be a vertex of P, we show that for all $\gamma \geq 0$ the point $x' = x + \gamma c$ belongs to P. Indeed, we have

$$x' = x + \gamma c = 1 \cdot x + \gamma \sum_{r \in R_p} \beta_r r = 1 \cdot x + \sum_{r \in R_p} \gamma \beta_r r.$$

Therefore, $x' \in P$, i.e. P is not bounded w.r.t. F, contradicting the hypothesis.

[$\Leftarrow$] Assume by contradiction that $c \in F \cap O_P$. By the decomposition theorem for convex polyhedra [Schrijver 1986], since O_P is the characteristic cone of P, there exists a non-empty convex polyhedron P' such that $P = P' \oplus O_P$. In particular, as $\mathbf{0} \in O_P$, we have that P' is a subset of P. Moreover, since $c \in O_P$, also $\delta c \in O_P$ for all $\delta \geq 0$. We can then conclude that for all $p' \in P'$, it holds $p' + \delta c \in P$ for all $\delta \geq 0$. Therefore, P is not bounded w.r.t. $\{c\}$ and *a fortiori* w.r.t. F. ∎

[5]To ensure this condition, a duplicate generator $x \in V \cap C$ can be removed from C, while a duplicate generator $x \in R \cap (V \cup C)$ can be replaced in R by a scalar multiple δx, for $\delta > 0$, that does not belong to $V \cup C$.

6. COMPUTING A SUITABLE OVER-APPROXIMATION

Theorem 4.5 leaves us with one problem: We need to compute a polyhedron $Over$ satisfying the assumptions of the theorem. As before, if not explicitly stated, we consider a fixed location l with a closed and bounded convex polyhedron F representing the flow constraint, and we omit the notations l and F whenever possible.

The first result states that the set $t\text{-}bnd(G)$ of points that are t-bounded in G can easily be computed by collecting those polyhedra in $[\![G]\!]$ that are bounded w.r.t. F.

THEOREM 6.1. *Given a polyhedron G, let B be the subset of $[\![G]\!]$ containing the convex polyhedra that are bounded w.r.t. F. Then, $t\text{-}bnd(G) = \bigcup_{P \in B} P$.*

PROOF. Let us consider the two inclusions separately.
($\supseteq$) This is immediate by observing that any polyhedron in B is bounded w.r.t. F and, by Corollary 5.6, t-bounded w.r.t. F as well. Hence, if $p \in \bigcup_{P \in B} P$ then $p \in t\text{-}bnd(G)$.
($\subseteq$) Let $p \in t\text{-}bnd(G)$. Then there must be at least one convex polyhedron $P \in [\![G]\!]$ with $p \in P$. If P is bounded w.r.t. F, then $P \in B$ and the thesis follows.

If, on the other hand, P is not bounded w.r.t. F, then there exists a point $p' \in P$ and a slope $c \in F$ such that $p' + c\,\delta \in P$, for all $\delta \geq 0$. Hence, c is a direction of infinity of P and, being P a convex polyhedron, the same property holds for all the points in P including p, therefore $p + \delta c \in P$, for all $\delta \geq 0$. This contradicts the hypothesis that $p \in t\text{-}bnd(G)$. Hence the conclusion. ∎

The effective computation of the operator $t\text{-}bnd(G)$ is, then, ensured by Theorem 5.8. In the following, we present two different over-approximations that satisfy the assumptions of Theorem 4.5.

The first over-approximation. Given two disjoint polyhedra U and V, let

$$Over_1 = U \cup t\text{-}bnd(\overline{U} \cap \overline{V}).$$

We prove that $Over_1$ satisfies the two assumptions of Theorem 4.5. Theorem 6.1 ensures that $Over_1 \setminus U$ is t-bounded. The following lemma proves the other assumption.

LEMMA 6.2. *It holds $RWA^{\mathrm{M}}(U, V) \subseteq Over_1 \subseteq \overline{V}$.*

PROOF. For the first inclusion, let $u \in RWA^{\mathrm{M}}(U, V)$. If $u \in U$, the thesis is obvious. Otherwise, $u \in \overline{U}$ and, by definition of RWA^{M}, $u \in \overline{V}$: hence, $u \in \overline{U} \cap \overline{V}$. Moreover, for all trajectories $f \in Adm(\langle l, u\rangle)$ there exists $\delta \geq 0$ such that $f(\delta) \in U$. Hence, u is t-bounded in $\overline{U} \cap \overline{V}$. By Item *(ii)* of Theorem 6.1, $u \in t\text{-}bnd(\overline{U} \cap \overline{V}) \subseteq Over_1$.

For the second inclusion, let $u \in Over_1$. If $u \in U$, clearly $u \notin V$. Otherwise, $u \in t\text{-}bnd(\overline{U} \cap \overline{V}) \subseteq \overline{U} \cap \overline{V} \subseteq \overline{V}$, and we are done. ∎

The second over-approximation. We propose an alternative over-approximation $Over_2$, which significantly improves the performance of computing RWA^{M}, as shown in Section 7. To this end, let us first introduce the following operator. Given a polyhedron G and a convex polyhedron F, the *positive pre-flow* operator $G \swarrow^{\exists}_{>0} F$ is defined as follows:

$$G \swarrow^{\exists}_{>0} F = \{u - \delta c \mid u \in G, c \in F, \delta > 0\}.$$

Intuitively, $G \swarrow^{\exists}_{>0} F$ contains the points that may reach G via a straight trajectory of non-zero length whose slope is in F. Notice that, for a convex polyhedron P, $P \swarrow^{\exists}_{>0} F$ is also a convex polyhedron. We write $G \swarrow^{\exists}_{>0}$ as an abbreviation for $G \swarrow^{\exists}_{>0} F$.

The following recent result shows how to efficiently compute, for convex polyhedra P and F, the forward version $P \nearrow_{>0} F$ of the operator above, called positive post-flow, by using the generator representation.

THEOREM 6.3. *[Benerecetti et al. 2011b] Given two convex polyhedra P and F, let (V_P, C_P, R_P) (resp., (V_F, C_F, R_F)) be a generator system for P (resp., F). The triple $(V_P \oplus V_F, C_P \cup V_P, R_P \cup V_F \cup C_F \cup R_F)$ is a generator system for $P \nearrow_{>0} F$.*

By observing that $P \swarrow^{\exists}_{>0} F = P \nearrow_{>0} -F$ and that the operator distributes over unions in its first argument, operator $G \swarrow^{\exists}_{>0} F$, for a general polyhedron G and a convex F, can easily be computed by exploiting the above theorem.

Let, now, $(V_F, \emptyset, \emptyset)$ be a generator system for F (being closed and bounded, F has no closure points and no rays), we define the following operator:

$$G \swarrow^{\mathrm{gen}} F \triangleq G \cup \bigcap_{g \in V_F} \big(G \swarrow^{\exists}_{>0} \{g\} \big).$$

Intuitively, $G \swarrow^{\mathrm{gen}} F$ contains the set of points from which the system reaches G along all the directions corresponding to the generators of F. The second over-approximation for RWA^{M} can then be defined as follows:

$$Over_2 = \Big(U \cup t\text{-}bnd\big((U \swarrow^{\mathrm{gen}} F) \setminus U \big) \Big) \setminus V.$$

Example 6.4. Consider the situation in Figure 9, where U and V are the two shaded

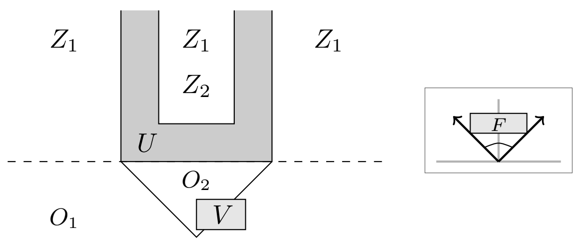


Fig. 9. An example showing the two over-approximations. Polyhedra U, O_1, Z_1, and Z_2 are unbounded and drawn truncated. It holds $Over_1 = U \cup O_1$ and $Over_2 = U \cup O_2$.

areas. In particular, U is an unbounded U-shaped non-convex polyhedron (drawn truncated). The first over-approximation consists in the points in U, plus the result of applying $t\text{-}bnd$ to $\overline{U} \cap \overline{V}$. The $t\text{-}bnd$ operator removes from its argument the three unbounded convex polyhedra denoted by Z_1. The result is $Over_1 = U \cup O_1$, where O_1 is the half-plane below the dashed line, excluding V.

For the second over-approximation, we first compute $(U \swarrow^{\mathrm{gen}} F) \setminus U$, which is the union of O_2 and Z_2. Then, the $t\text{-}bnd$ operator removes the polyhedron Z_2, because it is not bounded w.r.t. F (notice that $Z_2 \swarrow^{\exists}_{>0} = Z_2$). Hence, we obtain that $Over_2 = U \cup O_2$.

Once again, we prove that $Over_2$ satisfies the two assumptions of Theorem 4.5. Theorem 6.1 ensures that $Over_2 \setminus U$ is t-bounded. The following lemma proves the other assumption.

LEMMA 6.5. *It holds $RWA^{\mathrm{M}}(U, V) \subseteq Over_2 \subseteq \overline{V}$.*

PROOF. The fact that $Over_2 \subseteq \overline{V}$ is obvious by definition. Regarding the other inclusion, let $x \in RWA^{\mathrm{M}}(U, V)$. By definition, $x \notin V$, so we are left to prove that

$$x \in U \ \cup \ t\text{-}bnd\big((U \swarrow^{\mathrm{gen}} F) \setminus U \big).$$

If $x \in U$, the thesis is trivially true. Otherwise, we prove that $x \in t\text{-}bnd\big((U \swarrow^{\mathrm{gen}} F) \setminus U \big)$. To this purpose, we prove that $x \in (U \swarrow^{\mathrm{gen}} F) \setminus U$ and that x is t-bounded in it. By Item *(ii)* of Theorem 6.1, this implies the thesis.

Let $g \in V_F \subseteq F$ be any point of F. Since $x \notin U$ and $g \in F$, by definition of RWA^{M} it holds $x \in U \swarrow^{\exists}_{>0} \{g\}$. Finally, from $x \in (U \swarrow^{\mathrm{gen}} F) \setminus U$ and $x \in RWA^{\mathrm{M}}(U, V)$, we immediately obtain that for every $f \in Adm(x, F)$, there is $\delta > 0$ such that $f(\delta) \in U$. Since, however, U and $(U \swarrow^{\mathrm{gen}} F) \setminus U$ are disjoint, we conclude that $f(\delta) \notin (U \swarrow^{\mathrm{gen}} F) \setminus U$. Hence, x is t-bounded in $(U \swarrow^{\mathrm{gen}} F) \setminus U$. ∎

7. EXPERIMENTS WITH SPACEEX+

We implemented the algorithms described in the previous sections on top of the open-source tool SpaceEx [Frehse et al. 2011] [6]. In this section we show some results obtained by running the package against the maze example introduced in Section 2. The experiments were performed on an Intel Core i5-2400 (3.10GHz) PC.

We consider three versions of the maze example, which differ in their dynamics. In the first version, called *Det* and depicted in Figure 2, the vehicle follows exactly the current direction at constant speed, with no disturbances. Moreover, the vehicle speed combined with the mandatory delay between changes of direction ensure that U-turns are impossible. In the second version, called *Cyclic*, we decrease the mandatory delay to $\frac{1}{3}$, so that the vehicle is able to perform U-turns. Finally, in the third version, called *Non-Det*, the vehicle is subject to disturbances both in the direction of movement and laterally; U-turns are disabled.

The dynamics of the North direction in the three scenarios is described by the following table:

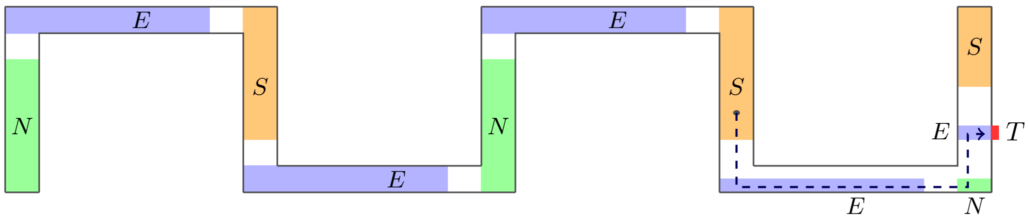
Version	$\dot{x}$	$\dot{y}$	$\dot{t}$
Det	0	2	1
Non-det	[-0.02, 0.02]	[1.95, 2.05]	1
Cyclic	0	2	3

We tested our implementation on progressively more complex mazes, by increasing the number of corridors. Figure 10 shows the shape of the longest maze (9 corridors) and the section of the winning regions for $t = 0$. The target is denoted by T and areas filled with the same color represent the winning region for a specific initial direction. Shorter mazes are obtained by progressively removing those corridors that are further away from the target.

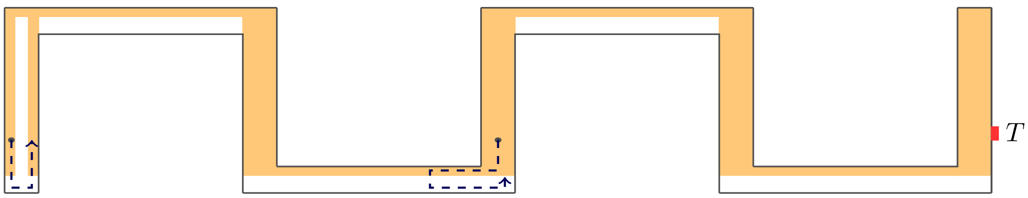
Consider Figure 10(b), which represents the winning region for the direction South. The first corridor on the left is split into three vertical stripes, of which only the middle one is not winning. Indeed, if the vehicle is moving down in the middle of the corridor, it is not able to perform a full U-turn without hitting the walls. On the contrary, the dashed trajectory on the bottom left corner of Figure 10(b) demonstrates a legal U-turn.

Next, let us focus on the 5 areas corresponding to location East in Figure 10(c), denoted by $E_1, \ldots, E_5$. Notice that the area labeled E_4 covers only half the width of the horizontal corridor. Indeed, if the vehicle is located in the other half of the corridor, when turning north it will be too close to the target and will not be able to take the second turn towards the target in time. The area E_3 ends 2 meters before the east wall, as beyond that the vehicle cannot avoid hitting the wall before being able to turn south. Finally, the points in the area E_5 are trivially winning, as they can reach the target by simply proceeding east. All areas E_i become gradually smaller as we move away from the target, due to the lateral uncertainty.

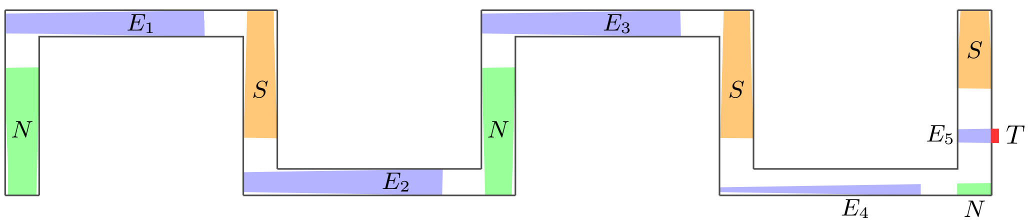
[6] A pre-release of our implementation, called SpaceEx+, can be downloaded at `http://wpage.unina.it/m.faella/spaceexplus`.



(a) Det. The winning region for the direction West coincides with the target.



(b) Cyclic, direction South.



(c) Non-Det. The winning region for the direction West coincides with the target.

Fig. 10. Winning regions for the three scenarios, when $t = 0$.

	Number of corridors							
Version	2	3	4	5	6	7	8	9
Det (O1)	0.40	0.45	0.72	1.05	2.10	2.62	3.84	5.65
Det (O2)	0.28	0.35	0.55	0.83	1.65	2.21	3.05	4.25
Cyclic (O1)	1.30	2.11	3.82	6.45	9.12	16.02	23.17	31.37
Cyclic (O2)	1.23	2.40	4.21	7.05	12.10	16.61	23.06	27.50
Non-Det (O1)	1.81	5.02	8.83	23.05	41.16	83.50	100.26	375.05
Non-Det (O2)	0.55	3.46	4.61	15.14	23.45	40.14	50.48	102.50

Fig. 11. Performances in seconds for the three scenarios, using the two overapproximations.

The table in Figure 11 shows the run time in seconds for the three different versions and an increasing number of corridors. As anticipated in the previous section, the results confirm that the second overapproximation $Over_2$ (indicated by O2 in the figure) improves the performances of the synthesis procedure in most cases, especially when complex dynamics is involved. Although still limited in scope, the results show that the proposed approach is practical, at least for relatively small problems.

8. CONCLUSIONS

In this paper we considered the problem of automatically synthesizing a switching controller for Linear Hybrid Games with respect to reachability objectives. The problem was considered in the literature only for decidable subclasses, such as Initialized Rectangular Hybrid Games [Henzinger et al. 1999]. We present a sound and complete symbolic algorithm for the finite-horizon case, based on the RWA^{M} operator. The main technical insight is that the related RWA^{m} operator can be used to compute RWA^{M}, by refining a suitable over-approximation.

We implemented the procedure in the tool SpaceEx and performed some preliminary experiments, showing that the procedure converges in non-trivial cases and that the approach is practical, at least for relatively small case studies.

The work presented in this paper paves the way for some interesting future work. For instance, we are currently investigating the problem of automatically constructing a concrete control strategy, which, coupled with the hybrid system, would result in a closed system, amenable to automatic verification by state-of-the-art analysis tools. The obtained closed system might be verified w.r.t. other properties of interest, such as stability, performance, etc.

Another interesting line of future investigation lies in selecting specific control strategies according to some optimality measure, such as reaching in minimum time or cost.

A. ADDITIONAL PROOFS FROM SECTION 3

We provide a proof of Lemma 3.3. Let $\{W_i\}_{i\geq 0}$ and $\{\sigma_i\}_{i\geq 0}$ be the sequence of winning regions and strategies defined in the proof of Theorem 3.1. The following Lemma holds.

LEMMA A.1. *For all $n \geq 0$, $s \in W_n$ if and only if for all runs $r \in Runs(s, \sigma_n)$, r reaches W_{n-1} within the first joint step.*

PROOF. [*if*] The thesis follows immediately from the definition of $CPre^{\mathrm{R}}(\cdot)$, as σ_n serves as a witness to $s \in CPre^{\mathrm{R}}(W_{n-1}) \subseteq W_n$.

[*only if*] If $s \in W_{n-1}$ the thesis is obviously true. Otherwise, $s \in W_n \setminus W_{n-1}$ and hence $s \in CPre^{\mathrm{R}}(W_{n-1})$. Assume by contradiction that there exists a run $r \in Runs(s, \sigma_n)$ such that its first joint step $s \xrightarrow{f,\delta} s' \xrightarrow{e} s''$ does not visit any state in W_{n-1}. Formally, for all $\delta' \in [0, \delta]$ we have $\langle loc(s), f(\delta')\rangle \notin W_{n-1}$ and moreover $s'' \notin W_{n-1}$ (Point *(i)*). We shall consider finite-duration joint steps, as infinite-duration steps can be treated similarly.

If $e \in Edg_{\mathrm{c}}$, by definition of consistency we have that $val(s') \otimes val(s'') \in \sigma_n(e)$. Since $s' \notin W_{n-1}$ and by the definition of σ_n, we have that $val(s') \otimes val(s'') \in Jump(W_n \setminus W_{n-1}, e, W_{n-1})$. On the other hand, $s'' \notin W_{n-1}$, which is a contradiction.

Hence, it must be $e \in Edg_{\mathrm{u}}$. Since, however, $s'' \notin W_{n-1}$ by Point *(i)*, we derive that $s' \notin CPre^{\mathrm{R}}(W_{n-1})$ and $s' \notin W_n$ (Point *(ii)*). We now distinguish two cases.

Case 1. No controllable transition leading to W_{n-1} is enabled during the timed step from s to s'. Then, for all strategies σ, either the joint step above is consistent with σ or the strategy interrupts the trajectory f before time δ, obtaining a different joint step of the form $s \xrightarrow{f,\delta'} t \xrightarrow{e'} t'$, where $e' \in Edg_{\mathrm{c}}$ and $t' \notin W_{n-1}$. In both cases, W_{n-1} is not reached within one joint step, contradicting $s \in CPre^{\mathrm{R}}(W_{n-1})$.

Case 2. Let $l = loc(s)$ and assume that there exists a delay $\delta' \in [0, \delta]$ such that in $\langle l, f(\delta')\rangle$ there is an enabled controllable transition that may lead to W_{n-1}. Let Δ be the set of all such δ' and let $\hat{\delta} = \inf \Delta$. Notice that $t \triangleq \langle l, f(\hat{\delta})\rangle \notin W_{n-1}$ by Point *(i)*.

Assume first that $\hat{\delta} \in \Delta$. It follows that the joint step of the form $r' = s \xrightarrow{f,\hat{\delta}} t \xrightarrow{e'} t'$, with $e' \in Edg_{\mathrm{c}}$ and $t' \in W_{n-1}$, is possible. If $t \in W_n \setminus W_{n-1}$, then $val(t) \otimes val(t') \in$

$P \triangleq \mathit{Jump}(W_n \setminus W_{n-1}, e', W_{n-1}) \subseteq \sigma_n(e')$. Then, by definition of consistency, f cannot exit from the activated region $P\downharpoonright_X$ without the system taking some discrete transition. Hence, for all $\delta'' \in [\hat{\delta}, \delta]$, it holds that $f(\delta'') \in P\downharpoonright_X \subseteq \sigma_n(e')\downharpoonright_X$ and, therefore, $\langle l, f(\delta'')\rangle \in W_n$. In particular, $s' \in W_n$, which contradicts Point *(ii)*. Hence, $t \notin W_n$ (Point *(iii)*), and, in general, it holds that $\langle l, f(\gamma)\rangle \notin W_n$, for all delays $\gamma \in \Delta$.

We can show that, for all strategies σ, there is a run from state t compatible with σ that does not reach W_{n-1} within the first joint step. Let σ be an arbitrary strategy. If σ allows some controllable transition to be taken before time $\hat{\delta}$, then by definition of $\hat{\delta}$, the transition leads outside W_{n-1}, contradicting $s \in \mathit{CPre}^{\mathrm{R}}(W_{n-1})$. If instead σ does not prescribe any controllable transition before time $\hat{\delta}$, the timed step $s \xrightarrow{f,\hat{\delta}} t$ is consistent with σ. Since $t \notin W_n$, by Point *(iii)*, and the discrete step $t \xrightarrow{e'} t'$ (i.e., the second step of r') leads to W_{n-1}, there must be another discrete step $t \xrightarrow{e''} t''$ such that $e'' \in \mathit{Edg}_{\mathrm{u}}$ and $t'' \notin W_{n-1}$. Hence, the sequence $s \xrightarrow{f,\hat{\delta}} t \xrightarrow{e''} t''$ is consistent with σ and contradicts $s \in \mathit{CPre}^{\mathrm{R}}(W_{n-1})$.

Next, assume $\hat{\delta} \notin \Delta$. For all $\epsilon > 0$ there exists $\epsilon' \in [0, \epsilon)$ s.t. $\hat{\delta} + \epsilon' \in \Delta$ and, hence, $\langle l, f(\hat{\delta}+\epsilon')\rangle \xrightarrow{e_1} t_1$, with $e_1 \in \mathit{Edg}_{\mathrm{c}}$ and $t_1 \in W_{n-1}$. We proved above that $\langle l, f(\hat{\delta}+\epsilon')\rangle \notin W_n$ and, as a consequence, the steps $\langle l, f(\hat{\delta} + \epsilon')\rangle \xrightarrow{e_2} t_2$, where $e_2 \in \mathit{Edg}_{\mathrm{u}}$ and $t_2 \notin W_{n-1}$, are also possible. Since there are only finitely many uncontrollable transitions, there exists an uncontrollable transition e_u leading outside W_{n-1} that is enabled along f in infinitely many points, arbitrarily close to $f(\hat{\delta})$. Formally, for all $\epsilon > 0$, there exists $\gamma_\epsilon \in [0, \epsilon)$ s.t. $\hat{\delta} + \gamma_\epsilon \in \Delta$ and $\langle l, f(\hat{\delta} + \gamma_\epsilon)\rangle \xrightarrow{e_u} t_u$, where $t_u \notin W_{n-1}$. Then, there exists a convex polyhedron $B \in [\![\mathit{Pre}(e_u, \overline{W_{n-1}})]\!]$ that is adjacent to the point $f(\hat{\delta})$ and contains the points $f(\hat{\delta} + \gamma_\epsilon)$, for infinitely many values of ϵ. Let $\bar{s}$ be a point in B such that $\bar{s} = \langle l, f(\hat{\delta} + \bar{\epsilon})\rangle$ for some $\bar{\epsilon} > 0$. Let f' be the straight-line trajectory going from $f(\hat{\delta})$ to $f(\hat{\delta} + \bar{\epsilon})$ in time $\bar{\epsilon}$. Formally, $f'(\delta') = f(\hat{\delta}) + \frac{f(\hat{\delta}+\bar{\epsilon})-f(\hat{\delta})}{\bar{\epsilon}} \cdot \delta'$ for all $\delta' \in [0, \bar{\epsilon}]$, which is admissible by Lemma 3.2. Since $f(\hat{\delta}) \in \mathit{cl}(B)$ and $f(\hat{\delta} + \bar{\epsilon}) \in B$, then the straight line f' is completely contained in B, by convexity of B. Let σ be an arbitrary strategy and

$$E_{f'} = \{e \in \mathit{Edg}_{\mathrm{c}} \mid \forall \delta' > 0\ \exists \delta'' \in [0, \delta')\ .\ f'(\delta'') \in \sigma(e)\downharpoonright_X\}$$

be the set of controllable transitions enabled by σ at times arbitrarily close to 0 along f'. Consider the case where $E_{f'}$ is not empty. Since, for every $e \in \mathit{Edg}_{\mathrm{c}}$, $\sigma(e)$ is a polyhedron, then for each $e \in E_{f'}$ there must be a convex polyhedron $P_e \in [\![\sigma(e)\downharpoonright_X]\!]$ adjacent to $f'(0)$ and some $\gamma_e \in (0, \epsilon]$ such that $f'(\gamma) \in P_e$, for all $\gamma \in (0, \gamma_e)$. Let $Y = \bigcap_{e \in E_{f'}} P_e$ and $\Gamma = \{\gamma \in (0, \bar{\epsilon}] \mid f'(\gamma) \in Y\}$. Since f' is a straight-line trajectory and Y is a non-empty convex polyhedron adjacent to $f'(0)$ containing an initial segment of f' (excluding the point $f'(0)$), Γ contains a left-open interval with infimum 0 and a positive supremum γ^*. Therefore, the sequence of steps $r'' = s \xrightarrow{f,\hat{\delta}} t \xrightarrow{f',\gamma^*/2} t' \xrightarrow{e} t''$, where $e \in \mathit{Edg}_{\mathrm{u}}$ and $t'' \notin W_{n-1}$, is consistent with σ and contradicts $s \in \mathit{CPre}^{\mathrm{R}}(W_{n-1})$.

Finally, assume that $E_{f'} = \emptyset$ and let Γ' be the set of time instants γ' such that no controllable transition is enabled by σ in $f'(\gamma')$. Since $E_{f'} = \emptyset$, Γ' is not empty and contains an interval with infimum 0 and positive supremum γ^*. Then, a sequence of steps of the same form as r'' is consistent with σ, contradicting $s \in \mathit{CPre}^{\mathrm{R}}(W_{n-1})$. Hence, all runs consistent with σ_n reach W_{n-1} in a single joint step. ∎

LEMMA A.2. *For all $n \geq 0$, states $s \in W_n$, and timed steps $s \xrightarrow{f,\delta} s'$ consistent with σ_n, either $s' \in W_n$ or the timed step encounters T.*

PROOF. We proceed by induction on n, with the case $n = 0$ being trivial. By contradiction, assume that $s' \notin W_n$ and T is not encountered during the timed step. By Lemma A.1 (*only if*), each run starting from s and consistent with σ_n reaches W_{n-1} within the first joint step. We distinguish two cases, both leading to contradictions.
Case 1. W_{n-1} is reached during the above timed step, i.e., $\bar{s} = \langle loc(s), f(\bar{\delta})\rangle \in W_{n-1}$ for some $\bar{\delta} \in [0, \delta]$. By the induction hypothesis applied to the residual timed step $\bar{s} \xrightarrow{\bar{f}, \delta-\bar{\delta}} s'$ (where $\bar{f}(\gamma) = f(\bar{\delta} + \gamma)$), either $s' \in W_{n-1} \subseteq W_n$ or T is encountered during the residual timed step. Both conclusions contradict our previous assumptions.
Case 2. W_{n-1} is not reached during the above timed step. Then, each joint step starting from s' and consistent with σ_n will reach W_{n-1}, because such step can be appended to $s \xrightarrow{f,\delta} s'$, giving rise to a joint step starting from s. By Lemma A.1 (*if*), we have $s' \in W_n$, which is a contradiction. ∎

LEMMA A.3. *For all $n \geq 0$ and $s \in W_n$, each joint step starting from s and consistent with σ_{n+1} is consistent with σ_n until it (possibly) reaches T.*

PROOF. Let $s \xrightarrow{\delta,f} s' \xrightarrow{e} s''$ be a joint step consistent with σ_{n+1} (the case of an infinite-duration step is analogous). The timed step from s to s' is consistent with σ_n because σ_n by definition contains fewer activation regions than σ_{n+1}, i.e., σ_n activates each controllable transition in a smaller region of the state-space. If the timed step from s to s' encounters T, we are done. Otherwise, by Lemma A.2 we have that $s' \in W_n$. If $e \in Edg_c$, we have that $val(s') \otimes val(s'') \in P$ for some $P \in \sigma_{n+1}(e)$. As by definition $\sigma_{n+1}(e) = \sigma_n(e) \cup \{Q\}$, where $Q \subseteq (W_{n+1} \setminus W_n) \otimes W_n$, it follows that $P \neq Q$ and hence $P \in \sigma_n(e)$ and the step $s' \xrightarrow{e} s''$ is consistent with σ_n. Finally, assume that $e \in Edg_u$. Then, the step $s' \xrightarrow{e} s''$ is obviously consistent with σ_n because σ_n cannot prevent the occurrence of uncontrollable transitions. ∎

LEMMA 3.3. *For all $n \geq 0$ and states $s \in W_n$, all runs starting from s and consistent with σ_n reach T.*

PROOF. The proof is by induction on n. The thesis is trivial for $n = 0$, since $W_0 = T$. Let $n > 0$ and $r \in Runs(s, \sigma_n)$. By Lemma A.1, r reaches a state $s' \in W_{n-1}$ within the first joint step. By Lemma A.3, the suffix of r starting from s' is consistent with σ_{n-1} until it (possibly) reaches T. By inductive hypothesis, every run starting from s' and consistent with σ_{n-1} eventually reaches T. So, r reaches T as required. ∎

ACKNOWLEDGMENT

The second author was partially supported by the Istituto Nazionale di Alta Matematica "F. Severi" (INdAM).

REFERENCES

R. Alur, T.A. Henzinger, and P.-H. Ho. 1996. Automatic Symbolic Verification of Embedded Systems. *IEEE Trans. Softw. Eng.* 22 (March 1996), 181–201. Issue 3. DOI:http://dx.doi.org/10.1109/32.489079

E. Asarin, O. Bournez, T. Dang, O. Maler, and A. Pnueli. 2000. Effective synthesis of switching controllers for linear systems. *Proc. IEEE* 88, 7 (2000), 1011–1025. DOI:http://dx.doi.org/10.1109/5.871306

R. Bagnara, P. M. Hill, and E. Zaffanella. 2008. The Parma Polyhedra Library: Toward a Complete Set of Numerical Abstractions for the Analysis and Verification of Hardware and Software Systems. *Science of Computer Programming* 72, 1–2 (2008), 3–21. DOI:http://dx.doi.org/10.1016/j.scico.2007.08.001

A. Balluchi, L. Benvenuti, T. Villa, H. Wong-Toi, and A. Sangiovanni-Vincentelli. 2003. Controller synthesis for hybrid systems with a lower bound on event separation. *Int. J. of Control* 76, 12 (2003), 1171–1200. DOI:http://dx.doi.org/10.1080/0020717031000123616

M. Benerecetti and M. Faella. 2013. Tracking Differentiable Trajectories across Polyhedra Boundaries. In *HSCC 13: Hybrid Systems Computation and Control. 16th Int. Conf.* ACM, 193–202. DOI:http://dx.doi.org/10.1145/2461328.2461360

M. Benerecetti, M. Faella, and S. Minopoli. 2011a. Revisiting Synthesis of Switching Controllers for Linear Hybrid Systems. In *Proc. of the 50th IEEE Conf. on Decision and Control*. IEEE.

M. Benerecetti, M. Faella, and S. Minopoli. 2011b. Towards Efficient Exact Synthesis for Linear Hybrid Systems. In *GandALF 11: 2nd Int. Symp. on Games, Automata, Logics and Formal Verification (Electronic Proceedings in Theoretical Computer Science)*, Vol. 54. DOI:http://dx.doi.org/10.4204/EPTCS.54.19

M. Benerecetti, M. Faella, and S. Minopoli. 2012. Reachability Games for Linear Hybrid Systems. In *HSCC 12: Hybrid Systems Computation and Control. 15th Int. Conf.* ACM, 65–74.

M. Benerecetti, M. Faella, and S. Minopoli. 2013. Automatic Synthesis of Switching Controllers for Linear Hybrid Systems: Safety Control. *Theoretical Computer Science* 493 (2013), 116–138.

P. Bouyer, T. Brihaye, and F. Chevalier. 2010. O-Minimal Hybrid Reachability Games. *Logical Methods in Computer Science* 6 (2010). Issue 1.

N. V. Chernikova. 1968. Algorithm for discovering the set of all the solutions of a linear programming problem. *U. S. S. R. Comput. Math. and Math. Phys.* 8, 6 (1968), 282–293. DOI:http://dx.doi.org/10.1016/0041-5553(68)90115-8

L. de Alfaro, M. Faella, T.A. Henzinger, R. Majumdar, and M. Stoelinga. 2003. The Element of Surprise in Timed Games. In *CONCUR 03: Concurrency Theory. 14th Int. Conf. (Lect. Notes in Comp. Sci.)*, Vol. 2761. Springer, 144–158. DOI:http://dx.doi.org/10.1007/978-3-540-45187-7_9

L. de Alfaro, T.A. Henzinger, and R. Majumdar. 2001. Symbolic Algorithms for Infinite-State Games. In *CONCUR 01: Concurrency Theory. 12th Int. Conf. (Lect. Notes in Comp. Sci.)*. Springer. DOI:http://dx.doi.org/10.1007/3-540-44685-0_36

G. Frehse, C. Le Guernic, A. Donzé, S. Cotton, R. Ray, O. Lebeltel, R. Ripado, A. Girard, T. Dang, and O. Maler. 2011. SpaceEx: Scalable Verification of Hybrid Systems. In *CAV 11: Proc. of 23rd Conf. on Computer Aided Verification*. 379–395.

L. C. G. J. M. Habets, P. J. Collins, and J. H. van Schuppen. 2006. Reachability and control synthesis for piecewise-affine hybrid systems on simplices. *IEEE Trans. on Automatic Control* 51, 6 (June 2006), 938–948. DOI:http://dx.doi.org/10.1109/TAC.2006.876952

T.A. Henzinger. 1996. The Theory of Hybrid Automata. In *11th IEEE Symp. Logic in Comp. Sci.* 278–292. DOI:http://dx.doi.org/0.1109/LICS.1996.561342

T.A. Henzinger, B. Horowitz, and R. Majumdar. 1999. Rectangular Hybrid Games. In *CONCUR 99: Concurrency Theory. 10th Int. Conf. (Lect. Notes in Comp. Sci.)*, Vol. 1664. Springer, 320–335. DOI:http://dx.doi.org/10.1007/3-540-48320-9_23

T.A. Henzinger, P.W. Kopke, A. Puri, and P. Varaiya. 1995. What's decidable about hybrid automata?. In *Proc. of the 27th annual ACM symposium on Theory of computing (STOC '95)*. ACM, 373–382.

T.A. Henzinger, P.W. Kopke, A. Puri, and P. Varaiya. 1998. What's Decidable about Hybrid Automata? *J. of Computer and System Sciences* 57, 1 (1998), 94 – 124. DOI:http://dx.doi.org/10.1006/jcss.1998.1581

Z. Lin and M.E. Broucke. 2006. Resolving Control to Facet Problems for Affine Hypersurface Systems on Simplices. In *Decision and Control, 2006 45th IEEE Conference on*. 2625–2630. DOI:http://dx.doi.org/10.1109/CDC.2006.377067

J. Lygeros, C. Tomlin, and S. Sastry. 1999. Controllers for reachability specifications for hybrid systems. *Automatica* 35, 3 (1999), 349 – 370. DOI:http://dx.doi.org/10.1016/S0005-1098(98)00193-9

O. Maler. 2002. Control from computer science. *Annual Reviews in Control* 26, 2 (2002), 175–187. DOI:http://dx.doi.org/10.1016/S1367-5788(02)00030-5

O. Maler, A. Pnueli, and J. Sifakis. 1995. On the Synthesis of Discrete Controllers for Timed Systems. In *12th Annual Symp. on Theor. Asp. of Comp. Sci. (Lect. Notes in Comp. Sci.)*, Vol. 900. Springer. DOI:http://dx.doi.org/10.1007/3-540-59042-0_76

M.L. Minsky. 1967. *Computation: Finite and Infinite Machines*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

P.J. Ramadge and W.M. Wonham. 1987. Supervisory Control of a Class of Discrete-Event Processes. *SIAM Journal of Control and Optimization* 25 (1987), 206–230. DOI:http://dx.doi.org/10.1137/0325013

A. Schrijver. 1986. *Theory of linear and integer programming*. John Wiley and Sons.

C.J. Tomlin, J. Lygeros, and S. Shankar Sastry. 2000. A game theoretic approach to controller design for hybrid systems. *Proc. of the IEEE* 88, 7 (2000), 949–970.

H. Le Verge. 1992. *A note on Chernikova's Algorithm*. Technical Report 635. IRISA, Rennes.

H. Wong-Toi. 1997. The synthesis of controllers for linear hybrid automata. In *36th IEEE Conf. on Decision and Control*. IEEE, San Diego, CA, 4607 – 4612. DOI:http://dx.doi.org/10.1109/CDC.1997.649708