

# Verifying Linear Temporal Properties on Polyhedral Systems: Decidability and Symbolic Algorithms

Massimo Benerecetti, Marco Faella, Fabio Mogavero

*Università degli Studi di Napoli Federico II, Italy*

*Gruppo Nazionale Calcolo Scientifico – Istituto Nazionale di Alta Matematica, Italy*

---

## Abstract

We study the problem of model checking linear temporal logic formulae on finite trajectories generated by polyhedral differential inclusions, thus enriching the landscape of models where such specifications can be effectively verified. Each model in the class comprises a static and a dynamic component. The static component features a finite set of observables represented by (non-necessarily convex) polyhedra. The dynamic one is given by a convex polyhedron constraining the dynamics of the system, by specifying the possible slopes of the trajectories in each time instant. We devise an exact algorithm that computes a symbolic representation of the region of points that existentially satisfy a given formula  $\varphi$ , *i.e.*, the points from which there exists a trajectory satisfying  $\varphi$ .

---

## 1. Introduction

Formal verification has been a central topic in computer science for decades, and model checking has emerged as a key technique for this purpose. In this paper, we focus on *continuous-time* and *infinite-state* systems, which are essential for cyber-physical applications [1]. We represent the state of our systems using a vector of real-valued variables, whose dynamics are governed by a constant polyhedral inclusion of the type  $\dot{x} \in F$ , where  $F$  is a convex polyhedron. Such dynamics correspond to the single-location dynamics of linear hybrid automata (LHAs) [2]. Whereas reachability in LHAs is undecidable [3], we show in this paper that model checking a linear temporal property on a single location is a decidable, albeit challenging, problem.

As specification language, we consider a real-time interpretation of linear temporal logic (LTL), called RTL following Reynolds [4]. In our interpretation, time is real-valued and each atomic proposition denotes a polyhedral region of the state-space. Hence, users can exploit the familiar syntax of LTL to express complex properties involving continuous variables and their relationships.

The polyhedral inclusions that define our trajectories bestow a considerable degree of flexibility, affording room for behaviours, commonly referred to as

*Zeno behaviours*, which may lack a plausible physical rationale or clash with the symbolic abstraction adopted in this paper. To avoid these issues, since our observables are polyhedral regions of the state-space, we restrict our attention to trajectories that transition between polyhedra finitely many times within any bounded time interval. We call this notion *well-behavedness* and compare it with similar notions in the existing literature.

We consider the problem of verifying an RTL formula  $\varphi$  against a *polyhedral system*, comprised of a flow constraint, which provides the specification of the polyhedral inclusions for the derivative of the system trajectories, and a polyhedral interpretation for all the atomic propositions appearing in the formulae. Given an RTL formula, a polyhedral system, and an initial point  $p \in \mathbb{R}^n$ , the model-checking problem asks whether there is a system trajectory that starts from  $p$  and satisfies the formula. The results of the model-checking problem can be used in two ways, depending on the interpretation given to the input model. Indeed, the non-determinism inherent in a polyhedral differential inclusion can be meant either in an angelic (*i.e.*, controllable) or demonic (*i.e.*, uncontrollable) sense. In the first case, a constraint of the type  $\dot{x} \in [1, 2]$  is taken to mean that the variable  $x$  can be steered by the system to grow with any rate between 1 and 2. In the second case, the same constraint signals that the environment may choose any growth rate between 1 and 2. Given a model with angelic non-determinism, one may use the results in this paper to verify that the system can be controlled into satisfying a specified property. If instead the non-determinism is meant to be interpreted as demonic, one will specify an error condition and check from which states the environment can generate a trajectory that engenders the error.

In fact, rather than solving the model-checking problem for a fixed initial point, we compute a symbolic representation of the set of initial points from which a trajectory satisfying the formula starts, a problem that we call the *RTL existential denotation problem*. The symbolic algorithm is based on a translation from RTL to LTL, followed by the classical automata construction for LTL [5, 6]. The automaton is then intersected with a suitable finite-state abstraction of the polyhedral system. We identify various classes of trajectories and formulae on which the existential denotation problem is decidable thanks to suitable discretisations of the polyhedral system.

Additionally, we present an on-the-fly symbolic algorithm that avoids the explicit product construction between the formula automaton and the discrete abstraction. We developed a prototype implementation of the on-the-fly algorithm, using SPOT [7] as the LTL-to-automaton translator and Parma Polyhedra Library [8] as the library for symbolic manipulation of polyhedra. A set of experiments show that our approach is effective. Our work has potential applications in a variety of domains, including robotics and control systems, and offers new insights into the analysis of polyhedral systems.

*A motivating example.* Consider a system of two tanks connected through a pump and holding some liquid. An inlet pours liquid into the first tank at an uncertain and time-varying rate, which however is known to be constrained to

the interval  $[1, 2]$ . The pump transfers liquid from the first tank to the second one at a variable rate also constrained to  $[1, 2]$ . Finally, an outlet discharges liquid from the second tank at a varying rate within  $[0, 3]$ . If we represent the level in the first (*resp.*, second) tank with variable  $a$  (*resp.*,  $b$ ) and we add a clock  $t$  to measure the passage of time, the above constraints lead to the dynamic laws reported in Figure 1. Notice that to properly capture the behaviour of the system, it is necessary to prevent both the tank levels and time from taking negative values. This requirement is ensured by imposing the invariant constraint  $t, a, b \geq 0$ .

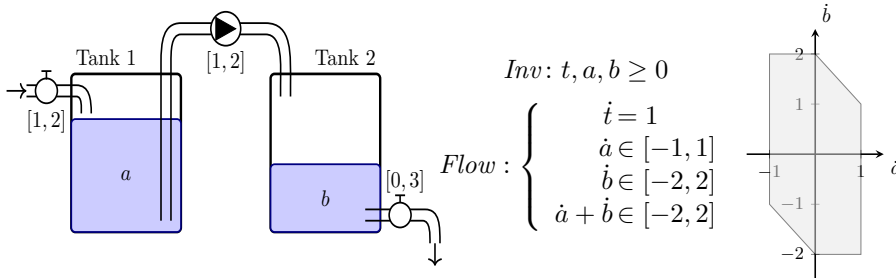


Figure 1: A system of two tanks. Left: A schematic representation. Right: The invariant, the flow, and the projection of the latter on the  $(\dot{a}, \dot{b})$  plane.

Suppose now that we want to compute the set of initial states from which the system, within the first 10 time units, can first reach a configuration where  $a \geq b + 1$ , and subsequently reach another configuration where  $b \geq a + 1$ . This property can be expressed by the following RTL formula:

$$\varphi_1^{\text{gap}} = (t = 0) \wedge \mathbf{G}(t \leq 10) \wedge \mathbf{F}(a \geq b + 1 \wedge \mathbf{F}(b \geq a + 1)).$$

This example also shows that, despite not directly supporting time bounds as parameter of the temporal operators, RTL still allows to reason about *absolute* time, by explicitly introducing an extra variable  $t$  into the model to represent it. In Section 9, we shall show how our algorithm can readily compute the set of initial points supporting a trajectory satisfying the above formula, as well as several variations thereof.

*Related work.* Several temporal logics have been proposed in the literature to express properties of real-time systems. Some proposals enrich classical temporal logic with new operators specific for real time, like decorating the *until* operator with time bounds. That is the case of MTL [9], MITL [10], and STL [11]. Other approaches, including ours, reinterpret the original LTL on real time. In particular, Reynolds investigates the validity problem for LTL interpreted over real time [4].

The dynamics we support generalise the single-mode (*i.e.*, single-location) dynamics of timed automata [12] and constant-rate multi-mode systems (MMS) [13], and correspond to the single-mode dynamics of linear hybrid automata

(LHA) [2]. In the case of MMS’s, reachability is a decidable problem, yet full LTL model checking is not. Notably, Blondin *et al.* have recently delineated a range of decidable syntactic fragments in this context [14]. When it comes to LHAs, even the reachability problem is undecidable [3]. This has not prevented the development of approximate or incomplete approaches, included in tools like SpaceEx [15] and NYCS [16].

If we go even higher in expressivity ladder of models for single-mode systems, the polyhedral inclusion characterising our model can be considered as a special case of an affine system with controllable inputs (*i.e.*, a dynamics of the type  $\dot{x} = Ax + b + Bu$  where  $A = 0$  and the control input  $u$  plays the role of nondeterminism). In that model, a sound but incomplete synthesis approach for LTL specifications was proposed [17].

Existing work [18, 19] addresses STL model-checking for general hybrid dynamics, but requires an *a priori* fixed time bound for trajectories. In contrast, we consider a more restricted form of dynamics, but support more general (and still decidable) semantics that include infinite-time trajectories and finite-time trajectories without a pre-defined time bound.

*Structure of the Paper.* The paper is organised as follows. Section 2 introduces polyhedral systems and their trajectories, considering both infinite-time and finite-time trajectories. For the latter ones, we also distinguish different termination criteria relative to the invariant constraint of the polyhedral system. We discussed the notion of well-behavedness and its relationship with other standard regularity conditions. Section 3 defines an abstract linear-time temporal language, called ALTL, that encompasses both classical (*i.e.*, discrete) LTL and its continuous version RTL. Here we also define the decision problem we are interested in solving, namely the computation of the existential denotation, which, in turn, allows us to solve the model checking problem for RTL *w.r.t.* a polyhedral system. Section 4 provides the technical framework to solve the decision problem, that allows to discretise trajectories into traces and RTL formulae into LTL formulae. Sections 5, 6 and 7, instead, present various finite abstractions, called polyhedral abstractions, and show how to solve the existential denotation problem for RTL formulae under each of the termination criteria identified in Section 3. Section 8 presents an efficient on-the-fly algorithm for computing the existential denotation for finite-time trajectories, and Section 9 describes the results of some experiments performed on a prototype implementation.

## 2. Polyhedral Systems

We study continuous-time and continuous-state dynamic systems, whose state  $x \in \mathbb{R}^n$  evolves non-deterministically under a differential inclusion constraint of the type  $\dot{x} \in Flow$ , for a fixed convex polyhedron  $Flow$ . In the following, we shall use the symbol  $\mathbb{R}_{\geq 0}$  to denote the set of non-negative reals and  $\bar{X}$  to denote the complement of a set  $X \subseteq \mathbb{R}^n$ .

*Polyhedra.* A *convex polyhedron* is the intersection of a finite number of strict or non-strict half-spaces. A *polyhedron* is a finite union of convex polyhedra and a *polytope* is a bounded convex polyhedron. We denote by  $Poly(\mathbb{R}^n)$  (*resp.*,  $CPoly(\mathbb{R}^n)$ ) the set of polyhedra (*resp.*, convex polyhedra) on  $\mathbb{R}^n$ . We generally use the letters  $P, Q$  to refer to convex polyhedra and letters  $A, B$  for general polyhedra, instead. For a polyhedron  $A$ , we denote by  $Patch(A)$  its representation as a finite set of convex polyhedra, called the *patches* of  $A$ . Moreover, we denote by  $cl(P)$  the *topological closure* of  $P$ , obtained by replacing all strict half-spaces with their non-strict version. Vice-versa, if all half-spaces are replaced by their strict version, we obtain the *interior* of  $P$ , denoted by  $int(P)$ . We say that a point  $x \in \mathbb{R}^n$  is *adjacent* to a polyhedron  $P$  if it belongs to its closure  $cl(P)$ . The *border* between polyhedra  $P$  and  $Q$  contains the points that are adjacent to both polyhedra and belong to at least one of them. Such a set corresponds to the polyhedron  $(P \cap cl(Q)) \cup (cl(P) \cap Q)$ . When the border between  $P$  and  $Q$  is not empty, we say that  $P$  and  $Q$  are *adjacent*.

*Polyhedral systems.* We are interested in dynamic systems that obey a given polyhedral differential inclusion. Assume first a finite set  $AP$  of atomic proposition symbols. Each atomic proposition  $p \in AP$  is interpreted as a polyhedron  $[p] \in Poly(\mathbb{R}^n)$ , called its *interpretation*, *i.e.*, the set of points where  $p$  holds. Then, consider a fixed convex polyhedron  $Flow \subseteq \mathbb{R}^n$  called the *flow constraint*, and we omit it from the notation whenever possible. We call the triple  $\mathcal{P} = (Flow, Inv, [\cdot])$  a *polyhedral system*, where  $Inv$  is a closed polyhedron representing the *invariant* region. For a set of atomic propositions  $\alpha \subseteq AP$ , we denote with  $\llbracket \alpha \rrbracket$  the interpretation of the set  $\alpha$ , namely the set of points where all and only the propositions in  $\alpha$  hold. Formally,

$$\llbracket \alpha \rrbracket = Inv \cap \bigcap_{p \in \alpha} [p] \cap \bigcap_{p \in AP \setminus \alpha} \overline{[p]}.$$

By definition,  $\llbracket \alpha \rrbracket$  is a polyhedron contained in the invariant. Observe that  $\llbracket \{p\} \rrbracket \subseteq [p]$  and the inclusion may be strict. Moreover, for any two sets of atomic propositions  $\alpha_1, \alpha_2 \subseteq AP$ , we have that either  $\llbracket \alpha_1 \rrbracket = \llbracket \alpha_2 \rrbracket$  or  $\llbracket \alpha_1 \rrbracket \cap \llbracket \alpha_2 \rrbracket = \emptyset$ . Hence, the image of  $2^{AP}$  under  $\llbracket \cdot \rrbracket$  is a partition of  $\mathbb{R}^n$  into polyhedra.

For real values  $a, b \in \mathbb{R}_{\geq 0}$ , we use the notation  $\langle a, b \rangle$  as a shorthand for one of the two types of right-closed intervals:  $(a, b]$  or  $[a, b]$ . For an interval  $I = \langle a, b \rangle$ , we denote by  $I_{>t}$  (*resp.*,  $I_{\geq t}$ ) the suffix interval  $(t, b]$  (*resp.*,  $[t, b]$ ), and similarly for the prefixes  $I_{<t}$  (*resp.*,  $I_{\leq t}$ ). Let  $\mathcal{Int}$  denote the set containing all the bounded right-closed intervals, that is the intervals  $I$  such that  $\sup(I) \in I$ , and the unbounded ones, namely those intervals whose supremum is equal to  $\infty$ .

Given a point  $x \in \mathbb{R}^n$  and an interval  $I \in \mathcal{Int}$ , a *trajectory* from  $x$  with time domain  $I$  is a function  $f: I \rightarrow \mathbb{R}^n$ , such that:

- (i)  $\lim_{t \rightarrow \inf(I)} f(t) = x$ ;
- (ii)  $f(t) \in Inv$ , for all  $t \in I$ ;

- (iii)  $f$  is continuous;
- (iv) whenever the derivative  $\dot{f}(t)$  is defined, it holds that  $\dot{f}(t) \in \text{Flow}$ ;
- (v)  $f$  is differentiable everywhere except for a finite number of points in any bounded interval of its domain.<sup>1</sup>

A trajectory is *finite-time* (*resp.*, *infinite-time*) if its time domain is bounded (*resp.*, unbounded). We shall use  $I_f$  to refer to the domain of the trajectory  $f$ . A trajectory  $f$  is *admissible in the polyhedral system*  $\mathcal{P}$  if it never exits the invariant, namely if  $f(t) \in \text{Inv}$ , for all  $t \in I_f$ . For a point  $x \in \mathbb{R}^n$ , let  $\text{Traj}(x)$  denote the set of trajectories admissible in  $\mathcal{P}$  starting from  $x$ . Given two trajectories  $f, f' \in \text{Traj}(x)$ , we say that  $f'$  is a *suffix* (*resp.*, a *prefix*) of  $f$  if  $f'$  is the restriction of  $f$  to a suffix (*resp.*, prefix) of its time domain, namely if  $I_{f'} = (I_f)_{\sim t}$  (*resp.*,  $I_{f'} = (I_f)_{\leq t}$ ), for some  $t \in I_f$  and  $\sim \in \{>, \geq\}$ . A trajectory is *well-behaved* if, in any bounded time interval  $\langle a, b \rangle \subseteq I$ , it crosses any hyperplane a finite number of times, *i.e.*, for all hyperplanes  $H$  there is a finite set of time instants  $a = t_0 < t_1 < \dots < t_k = b$  such that, during every open interval  $(t_i, t_{i+1})$ , the trajectory lies in the same closed half-space induced by  $H$ . Further properties of well-behaved trajectories are discussed in Section 2.1. We use  $\text{Traj}(x)$  to denote the set of all well-behaved trajectories starting from the given point  $x$ .

The following lemma is an adaptation of a result originally proved in [20] and ensures that if two points of a convex polyhedron can be connected by a trajectory admissible *w.r.t.* a convex flow constraint, then there is a straight admissible trajectory that connects the two points and requires the same time to do so. This result is used later on to prove results in Sections 2.2, 5 and 6.

**Lemma 1** ([20]). *For any convex flow constraint  $\text{Flow}$ , convex polyhedron  $X$  and points  $x_1, x_2 \in X$ , the following two conditions are equivalent for all  $t^* \geq 0$ :*

1. *there exists a trajectory  $f$  such that  $f(0) = x_1$  and  $f(t^*) = x_2$ ;*
2. *there is a straight-line trajectory  $f'(t) \triangleq x_1 + d \cdot t$ , with  $d \in \text{Flow}$ , such that  $f'(0) = x_1$ ,  $f'(t^*) = x_2$  and  $f'(t) \in X$ , for all  $t \in [0, t^*]$ .*

*Termination criteria of trajectories.* In the following, we adopt four semantics for a polyhedral system, depending on whether we are interested in finite or infinite well-behaved trajectories and on two different interpretations of the invariant. We first give an intuitive explanation of the four semantics and then define them formally:

[fin] **Finite-time.** This semantics considers only finite-time trajectories. It is suitable to properties that are positively verified as soon as a prefix of a trajectory satisfies them, such as reachability properties.

---

<sup>1</sup>Observe that this property is stronger than the classic notion of *differentiable almost everywhere*, since the latter allows the set of non-differentiable points to accumulate, while the former forbids such a possibility.

[inf] **Infinite-time.** This semantics considers only infinite-time trajectories. It is suitable to non-terminating properties, such as liveness or fairness properties.

[may] **Terminated by may-exit.** This semantics considers all trajectories that are either infinite-time, or end in a *may-exit* point, *i.e.*, a point that is on the boundary of the invariant and from which at least one admissible direction exits from the invariant. This semantics considers the invariant as the region of the state-space where the current dynamics hold. When reaching a may-exit point, the current dynamics may end because the system moves to another dynamics (not modelled in the current polyhedral system).<sup>2</sup>

[must] **Terminated by must-exit.** We consider all trajectories that are either infinite-time, or end in a *must-exit* point, *i.e.*, a point that is on the boundary of the invariant and from which all admissible directions exit from the invariant. This is a stronger interpretation of the invariant as an inescapable constraint on motion. Under this interpretation, a trajectory only ends when no dynamics are possible within the invariant.

The above semantics induce four sets of well-behaved trajectories starting from a given point, denoted respectively by  $Traj^{\text{fin}}$ ,  $Traj^{\text{inf}}$ ,  $Traj^{\text{may}}$ , and  $Traj^{\text{must}}$ . For a point  $x \in \mathbb{R}^n$ , the set  $Traj^{\text{fin}}(x)$  contains the admissible trajectories of  $\mathcal{P}$  with bounded domain, while  $Traj^{\text{inf}}(x)$  contains those with unbounded domain. In order to formalise precisely the sets  $Traj^{\text{may}}$  and  $Traj^{\text{must}}$ , we introduce the following notion. Given a polyhedron  $P$  and a trajectory  $f$ , we say that  $f$  *immediately enters*  $P$  if there exists a time  $t \in I_f \setminus \{\inf(I_f)\}$  such that  $f(t') \in P$ , for all  $\inf(I_f) < t' \leq t$ . Then, for a polyhedral system  $\mathcal{P}$  and a point  $x \in Inv$ , the following two equations capture precisely the informal descriptions of  $Traj^{\text{may}}$  and  $Traj^{\text{must}}$  given above:

$$Traj^{\text{may}}(x) = Traj^{\text{inf}}(x) \cup \left\{ f \in Traj^{\text{fin}}(x) \left| \begin{array}{l} \exists f' \in Traj_{\top}(f(\text{sup}(I_f))) \\ f' \text{ immediately enters } \overline{Inv} \end{array} \right. \right\},$$

$$Traj^{\text{must}}(x) = Traj^{\text{inf}}(x) \cup \left\{ f \in Traj^{\text{fin}}(x) \left| \begin{array}{l} \forall f' \in Traj_{\top}(f(\text{sup}(I_f))) \\ f' \text{ immediately enters } \overline{Inv} \end{array} \right. \right\}$$

where  $Traj_{\top}(x)$  denotes the set of trajectories from  $x$  admissible in the polyhedral system obtained from  $\mathcal{P}$  by replacing the invariant constraint of  $\mathcal{P}$  with the trivial one containing all the points in  $\mathbb{R}^n$ . The following inclusions hold between them and no other inclusions hold in general:

$$Traj^{\text{fin}}(x) \subseteq Traj(x)$$

$$Traj^{\text{inf}}(x) \subseteq Traj^{\text{must}}(x) \subseteq Traj^{\text{may}}(x) \subseteq Traj(x).$$

---

<sup>2</sup>That may be the case if we are verifying a local property of a single mode within a broader hybrid system.

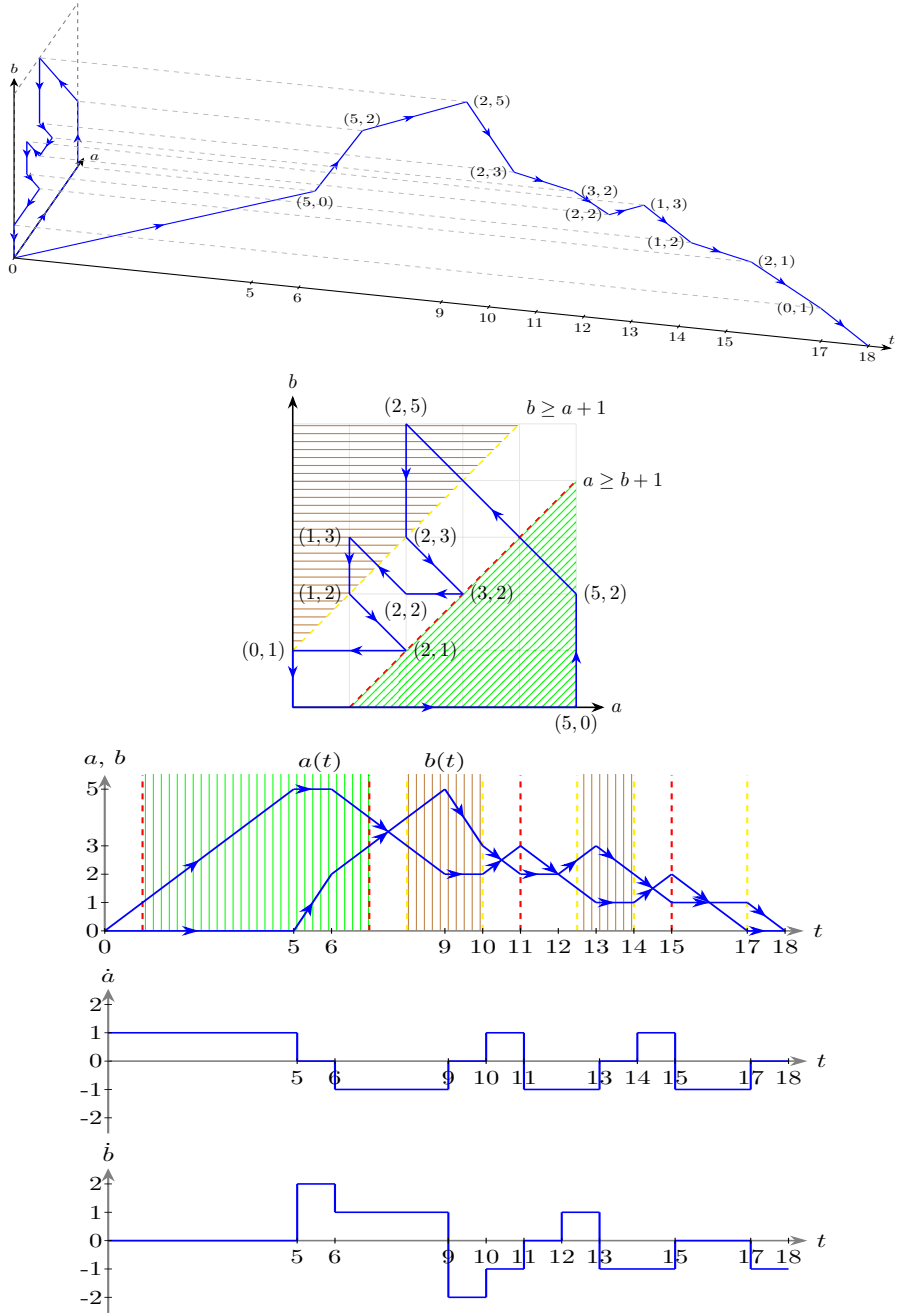


Figure 2: A finite-time trajectory of the two-tank system in the three-dimensional  $(t, a, b)$  space, along with its  $(a, b)$  projection, and the corresponding parametric representations of the values  $a(t)$  and  $b(t)$  and their derivatives  $\dot{a}(t)$  and  $\dot{b}(t)$ .

In Figure 2, we depict a finite-time trajectory for the two-tank system introduced earlier, *i.e.*, a trajectory in  $Traj^{\text{fin}}((0, 0, 0))$ , with a time span of 18 and time domain  $[0, 18]$ . This trajectory can be extended cyclically to an infinite one with period 18. Moreover, this also belongs to  $Traj^{\text{may}}((0, 0, 0))$ , since at its endpoint it can immediately exit the invariant set  $Inv = \{t, a, b \geq 0\}$  using, for example, the flow  $(\dot{a}, \dot{b}) = (-1, -1)$ . Finally,  $Traj^{\text{must}}(x)$  is empty, for any point  $x \in Inv$  of the invariant set, as it is always possible to remain inside it using a flow  $(\dot{a}, \dot{b})$  with non-negative components. Finally, one can easily verify that this trajectory fails to satisfy  $\varphi_1^{\text{gap}}$ , since it extends beyond time 10. In contrast, its prefix restricted to time 10 does satisfy the property, as it first enters the region  $a \geq b + 1$  and later transitions to  $b \geq a + 1$  within this time window.

The interpretation  $[\cdot]$  of the atomic propositions induces a mapping from trajectories to functions of type  $I \rightarrow 2^{AP}$ , called *signals* [21], over which we shall base the semantics of the logics defined in Section 3. Namely, given a trajectory  $f$ , we denote with  $\sigma_f$  the signal assigning to each time instant  $t \in I$  the set of atomic propositions that are true at  $f(t)$ . Formally,

$$\sigma_f(t) \triangleq \{p \in AP \mid f(t) \in [p]\}.$$

The notions of suffix, prefix and finite/infinite-time lift from trajectories to signals in the natural way.

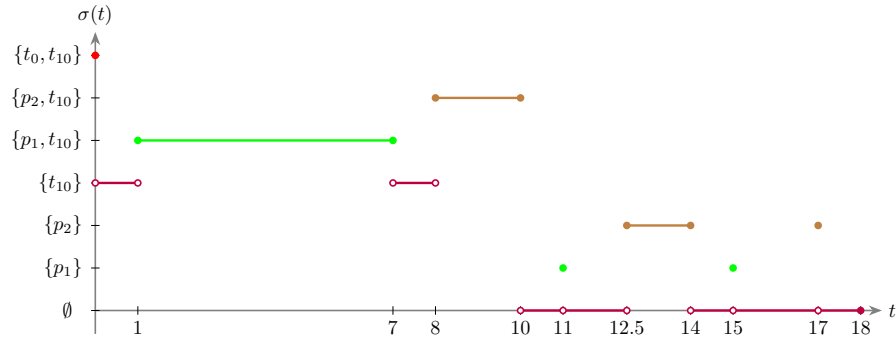


Figure 3: The finite-time signal  $\sigma : [0, 18] \rightarrow 2^{AP}$  corresponding to the finite-time trajectory depicted in Figure 2, where atomic propositions  $t_0$ ,  $t_{10}$ ,  $p_1$  and  $p_2$  represent the properties  $t = 0$ ,  $t \leq 10$ ,  $a \geq b + 1$ , and  $b \geq a + 1$ , respectively.

In Figure 3, we report an example of a finite-time signal representing the evolution of the properties a trajectory satisfies over time. The signal  $\sigma(t)$  actually corresponds to the finite-time trajectory of Figure 2 and takes values in  $\{\emptyset, \{t_{10}\}, \{p_1\}, \{p_2\}, \{t_0, t_{10}\}, \{p_1, t_{10}\}, \{p_2, t_{10}\}\} \subseteq 2^{AP}$ , where  $AP = \{t_0, t_{10}, p_1, p_2\}$  is the set of atomic propositions with  $[t_0] = \{(0, a, b) \in \mathbb{R}^3\}$ ,  $[t_{10}] = \{(t, a, b) \in \mathbb{R}^3 \mid t \leq 10\}$ ,  $[p_1] = \{(t, a, b) \in \mathbb{R}^3 \mid a \geq b + 1\}$ , and  $[p_2] = \{(t, a, b) \in \mathbb{R}^3 \mid b \geq a + 1\}$ . Dots indicate the value of the signal at crossing times: filled circles denote inclusion of the endpoint, while empty circles denote exclusion.

### 2.1. Well-Behavedness and Finite Variability

Considering the membership in a half-space as an observable, the well-behavedness condition previously introduced states that the truth value of the observable along the trajectory  $f$  changes only a finite number of times in every bounded time interval. This last property is equivalent to the notion of *discrete variation*, as observed in [22].

These notions can be compared to classical notions in analysis such as *analyticity* and *Lipschitz continuity*. Recall that a trajectory  $f$  is *analytic in a point*  $t$  in its domain if it is smooth at  $t$  and the Taylor's series of  $f$  at  $t$  converges to  $f$  in some open neighbourhood of  $t$ . Moreover,  $f$  is said to be *analytic* if it is analytic in every point of its domain and *piecewise analytic* if it is analytic in every point except for a finite number in any bounded interval of its domain. The following result follows from Theorem 16 of [22].

**Proposition 1.** *On the set of all trajectories, piecewise analyticity implies well-behavedness.*

A trajectory  $f$ , instead, is *Lipschitz continuous* on  $X \subseteq \mathbb{R}_{\geq 0}$  if there exists a constant  $K \geq 0$  such that, for all  $t_1, t_2 \in X$ , it holds that

$$\|f(t_1) - f(t_2)\| \leq K \cdot |t_1 - t_2|,$$

where  $\|\cdot\|$  denotes the Euclidean norm. Moreover,  $f$  is *locally Lipschitz continuous* if for all  $t \in \mathbb{R}_{\geq 0}$  there exists a neighbourhood of  $t$ , where  $f$  is Lipschitz continuous.

**Proposition 2.** *On the set of all trajectories, Lipschitz continuity and well-behavedness are incomparable notions.*

*Proof.* In  $\mathbb{R}^2$ , the trajectory  $f_1(t) = (t, t^2)$  is well-behaved but not Lipschitz continuous. Note that  $f_1$  is locally Lipschitz continuous. For the other non-implication, let  $f_2(0) = (0, 0)$ ,  $f_2(t) = (t, t^2 \cdot \sin(t^{-1}))$ , for all  $t \in (0, \pi^{-1})$ , and  $f_2(t) = (t, t - \pi^{-1})$ , for all  $t \geq \pi^{-1}$ . Then,  $f_2$  is Lipschitz continuous because it is differentiable in  $(0, +\infty)$  and its derivative is bounded. However, it is not well-behaved because it crosses the hyperplane  $y = 0$  infinitely often in any time interval  $(0, a)$ , with  $a > 0$ . ■

In Figure 4, we depict the  $b$ -projection of a finite-time Lipschitz-continuous but non-well-behaved trajectory  $f(t) = (t, a(t), b(t))$  for the two-tank system introduced earlier, where  $a(t) = t$  and  $b(t) = 1 + (\frac{t-1}{\pi})^2 \sin(\frac{\pi}{t-1})$ , for  $0 \leq t \leq 1$ .

We now provide an alternative characterisation of well-behavedness that is functional to the solution of our problem. Recall that a *polyhedral partition* of  $\mathbb{R}^n$  is a finite set of mutually disjoint convex polyhedra whose union is  $\mathbb{R}^n$ .

**Proposition 3.** *A trajectory is well-behaved iff, for all polyhedral partitions of  $\mathbb{R}^n$  and all time instants  $t \in \mathbb{R}_{\geq 0}$ , the trajectory changes polyhedron a finite number of times during  $[0, t]$ .*

*Proof.* Let us consider the two directions separately.

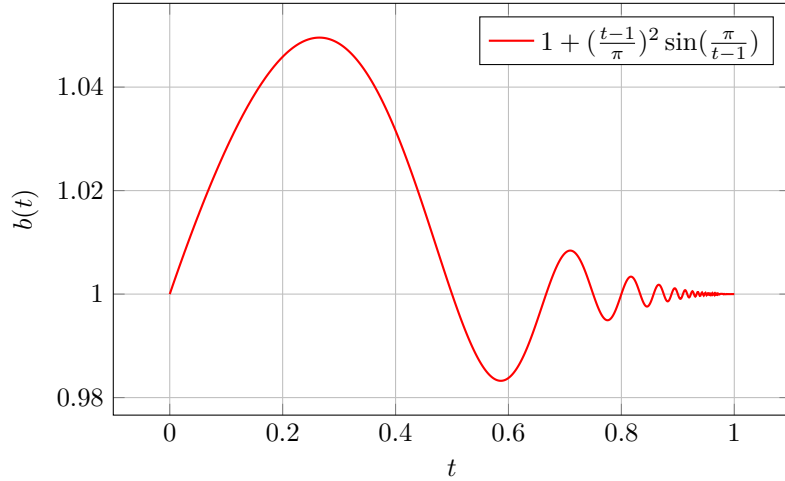


Figure 4: Projection of a finite-time Lipschitz continuous but non-well-behaved trajectory.

- **[Only-if direction].** Let  $f$  be a well-behaved trajectory and assume that there is a polyhedral partition  $\{P_1, \dots, P_k\}$  of  $\mathbb{R}^n$  such that  $f$  changes polyhedron an infinite number of times in the interval  $[0, t]$ , for some  $t \in \mathbb{R}^+$ . Then there is a convex polyhedron  $P_i$  that is entered and exited infinitely many times in the interval  $[0, t]$ . Since  $P_i$  is the intersection of a finite number of half-spaces of  $\mathbb{R}^n$ , there must be one such half-space that is entered and exited infinitely many times in the same interval. As a consequence, the hyperplane defining that half-space is traversed infinitely many times in the interval, contradicting the assumption of well-behavedness of  $f$ .
- **[If direction].** Assume  $f$  is not well-behaved. Then, by definition, there must be a time instant  $t \in \mathbb{R}^+$  and a hyperplane in  $\mathbb{R}^n$  that is traversed infinitely many times in the interval  $[0, t]$ . Take  $P$  as any one of the two half-spaces induced by the hyperplane. Clearly,  $P$  is a convex polyhedron which is entered and exited infinitely many times in  $[0, t]$ . Take, now, the polyhedral partition  $\{P, \bar{P}\}$  of  $\mathbb{R}^n$ . The trajectory  $f$  moves between  $P$  and  $\bar{P}$  infinitely many times in  $[0, t]$ , which gives us the thesis. ■

A *time slicing*  $\tau$  for an infinite-time signal  $\sigma: I \rightarrow 2^{A^P}$  is an enumerable increasing sequence of time points  $\{t_i\}_{i=0}^\omega \subseteq \mathbb{R}^+$ , with  $t_0 = \inf(I)$  and  $\sup\{t_i\}_{i=0}^\omega = \sup(I)$ , for which there is a corresponding sequence of observables  $\{\alpha_i\}_{i=0}^\omega \subseteq 2^{A^P}$  such that, for all indices  $0 \leq i < \omega$  and time instants  $t \in (t_i, t_{i+1})$ , it holds true that  $\sigma(t) = \alpha_i$ . *Mutatis mutandis*, a *time slicing*  $\{t_i\}_{i=0}^k \subseteq \mathbb{R}^+$  for a finite-time signal is defined accordingly. Observe that a time slicing necessarily contains a time point every time the original signal moves from one observable to another. In addition, it may also contain extra time points while staying in the same observable (*a.k.a. stuttering*). By  $TS(\sigma)$  we denote the set of all time slicings of  $\sigma$  and introduce  $TS(f) \triangleq TS(\sigma_f)$ , for any trajectory  $f$ .

Note that this set does not depend on whether a signal is left-open or not, *i.e.*,  $TS(\sigma) = TS(\sigma_{>\inf(I)})$ .

Consider again the signal shown in Figure 3. One possible time slicing is, *e.g.*,  $\tau_1 = \{0, 1, 7, 8, 10, 11, 12.5, 14, 15, 17, 18\}$ , while another is  $\tau_2 = \{0, 1, 4, 7, 8, 9, 10, 11, 12.5, 14, 15, 16, 17, 18\}$ . All valid time slicings for this signal, however, must include every point in  $\tau_1$ , as these correspond to the instants at which the signal changes its value.

We say that a signal  $\sigma: I \rightarrow 2^{AP}$  has *finite variability* if it changes its value only a finite number of times, in any bounded time interval  $\langle a, b \rangle \subseteq I$ , *i.e.*, if all finite-time sub-signals  $(\sigma_{\geq a})_{\leq b}$  admit a time slicing.

An immediate consequence of Proposition 3 is that for any polyhedral system  $\mathcal{P}$ , all well-behaved trajectories induce finite variability signals.

**Proposition 4.** *If a trajectory  $f$  is well-behaved, then the corresponding signal  $\sigma_f$  has finite variability.*

*Proof.* Take any polyhedral partitioning  $\{P_j\}_{j \in J}$  of  $\mathbb{R}^n$  that respects the propositions in  $AP$ , meaning that, for all  $j \in J$  and  $p \in AP$ , either  $P_j \cap [p] = \emptyset$  or  $P_j \subseteq [p]$ . Since  $f: I \rightarrow \mathbb{R}^n$  is well-behaved, it must change convex polyhedron in  $\{P_j\}_{j \in J}$  a finite number of times in any bounded time interval  $\langle a, b \rangle \subseteq I$ . Let  $P_{i_1}, \dots, P_{i_z}$  be the sequence of convex polyhedra traversed by  $f$  in that interval and  $\tau = \{t_i\}_{i=0}^k \subseteq \mathbb{R}^+$  the sequence of instants in which  $(f_{\geq a})_{\leq b}$  changes polyhedron in the sequence, with the possible addition of instants  $a$  and  $b$ , if needed. Since the polyhedral partitioning respects  $AP$ , every  $P_{j_i}$  is contained in  $[\alpha]$ , for some  $\alpha \subseteq AP$ . Hence,  $\tau$  is a suitable time-slicing of  $(f_{\geq a})_{\leq b}$ . The thesis follows from the arbitrariness of the interval extremes  $a$  and  $b$ . ■

It is interesting to observe that a property such as  $\mathbf{GF}(a \geq b + 1) \wedge \mathbf{GF}(b \geq a + 1)$  cannot be realised by finite-time well-behaved trajectories. Indeed, it can hold only for trajectories that either extend infinitely in time or exhibit infinitely many oscillations within a finite time interval, hence, not well-behaved.

Type	Name	Role	Symbol
$I \rightarrow \mathbb{R}^n$	Trajectory	Behaviour of a polyhedral system	$f$
$I \rightarrow 2^{AP}$	Signal	Interpretation of RTL	$\sigma$
$\{0, 1, \dots, k\} \rightarrow 2^{AP}$	Trace	Interpretation of LTL	$w$
$\{0, 1, \dots, k\} \rightarrow \mathbb{R}^+$	Time slicing	Time decomposition of a signal	$\tau$

Table 1: Notation used in the paper: three types of trace-like objects (from the most concrete to the most abstract) and the time decomposition of a signal.

In the rest of the paper we shall leave the polyhedral system  $\mathcal{P}$  implicit, consider only well-behaved trajectories and, as a consequence of Proposition 4, only finite variability signals.

## 2.2. Reachability Operators on Polyhedra

As an essential tool for the development of the solution of the model-checking problem considered in this work, we now introduce a geometric reachability

operator that works on polyhedra. The operator  $reach^b(A, B)$  takes as arguments two (possibly non-convex) polyhedra  $A$  and  $B$ , and collects the set of points of  $A$  that can reach  $B$ , following a trajectory that never leaves their union. The  $b$  superscript can be either 0 or +, corresponding to different timing constraints: a point from  $A$  belongs to  $reach^0(A, B)$  if it can *immediately* enter into  $B$ , whereas it belongs to  $reach^+(A, B)$  if it can enter into  $B$  after a strictly positive delay. Formally:

$$\begin{aligned} reach^0(A, B) &\triangleq \{x \in A \mid \exists f \in Traj(x), t > 0. \forall t' \in (0, t]. f(t') \in B\}; \\ reach^+(A, B) &\triangleq \{x \in A \mid \exists f \in Traj(x), t > 0. f(t) \in B \text{ and } \forall t' \in (0, t). f(t') \in A\}. \end{aligned}$$

Note that  $reach^b(A, B)$  and union operation commute on the second argument, but not on the first one, i.e.:

$$reach^b(A, B_1 \cup B_2) = reach^b(A, B_1) \cup reach^b(A, B_2). \quad (1)$$

It is sometimes useful to split the result of  $reach^b(A, B)$ , which is a general polyhedron, into convex polyhedra, each contained in one of the patches of  $A$ . To this aim, we introduce the following *split* function. For all polyhedra  $A$  and  $A' \subseteq A$ , the function  $split(A', A)$  returns a set of pairs  $\{(P_i, X_i)\}_{i=0}^n$  such that:

- (i)  $P_i$  and  $X_i$  are convex polyhedra such that  $X_i \subseteq P_i$ ,
- (ii) each  $P_i$  is one of the patches of  $A$ , and
- (iii)  $A'$  is the union of the  $X_i$ 's.

It is straightforward to implement the function *split* using Boolean operations on polyhedra.

*Computing the reach operators.* We now show how to compute the value of  $reach^b$  with a finite number of geometric operations, when its second argument is a *convex* polyhedron. When the second argument is not convex, one can resort to the distributivity property (1).

First, define the *positive pre-flow*  $P_{\prec_{>0}}$  of a convex polyhedron  $P$  as the set of points that can reach  $P$  after a strictly positive delay. Formally:

$$P_{\prec_{>0}} \triangleq \{x \in \mathbb{R}^n \mid \exists d \in F, t > 0. x + d \cdot t \in P\}.$$

The lemma below deals with  $reach^0$ , whereas the subsequent one provides an algorithm for  $reach^+$ . Their proofs can be found in [Appendix A.5](#).

**Lemma 2.** *For all polyhedra  $A$  and convex polyhedra  $B$ , the following holds:*

$$reach^0(A, B) = A \cap cl(B) \cap B_{\prec_{>0}}.$$

When it comes to computing  $reach^+$ , we shall make use of the *May Reach While Avoiding* operator  $RWA^m(Y, Z)$ , that collects the points from which some

admissible trajectory can reach a point in the set  $Y$ , while avoiding all the points in the set  $Z$ . The operator is formally defined as follows:

$$RWA^m(Y, Z) \triangleq \{x \in \mathbb{R}^n \mid \exists f \in \text{Traj}(x), t \geq 0. \\ f(t) \in Y \text{ and } \forall t' \in [0, t). f(t') \in Y \cup \overline{Z}\}.$$

An algorithm for computing  $RWA^m$  using symbolic operations on polyhedra has been presented in [23] and included in the tool NYCS<sup>3</sup>. The following lemma formalises the connection between  $RWA^m$  and  $\text{reach}^+$ .

**Lemma 3.** *For all polyhedra  $A$  and convex polyhedra  $B$ , the following holds:*

$$\text{reach}^+(A, B) = \bigcup_{P \in \text{Patch}(A)} RWA^m(T_P, \overline{A}), \quad \text{where } T_P \triangleq P \cap (\text{cl}(P) \cap B) \setminus \angle_{>0}.$$

As far as the *computational complexity* of  $\text{reach}^b$  is concerned, first notice that their computation is based on symbolic operations on polyhedra. All known algorithms underlying such computations are superpolynomial in the worst case [24]. A loose measure of complexity can be obtained by counting the number of symbolic operations involved. The operator  $\text{reach}^0$  involves a constant number of geometric operations, specifically intersections of polyhedra, closure operations and positive time-elapse [8]. The computation of  $\text{reach}^+(A, B)$ , instead, requires at most  $|\text{Patch}(A)|$  calls to  $RWA^m$ . An analysis of the algorithm for  $RWA^m$  (see Theorem 3 in [23]) shows that computing  $RWA^m(Y, \overline{Z})$  requires at most  $k \cdot m^{\alpha(m)}$  symbolic operations, where  $k$  and  $m$  are, respectively, the number of convex patches of  $Y$  and  $Z$ . The analysis also shows that the number of patches of the output cannot exceed  $m^{\alpha(m)}$ . Summarising,  $\text{reach}^+(A, B)$  requires at most  $m^{\alpha(m)}$  operations, with  $m$  the number of convex patches of  $A$ , since  $B$  is a single patch, and its output contains at most  $m^{\alpha(m)}$  patches.

### 3. Linear Temporal Logics

*Linear Temporal Logic* (LTL) was originally introduced by Pnueli [25] to specify and verify properties of reactive systems. Given a set of atomic propositions AP, an LTL formula is composed of atomic propositions, the Boolean connectives *negation* ( $\neg$ ), *conjunction* ( $\wedge$ ), and *disjunction* ( $\vee$ ), and the temporal operators *next* ( $\mathbf{X}$ ), *until* ( $\mathbf{U}$ ) and *release* ( $\mathbf{R}$ ).

LTL formulae are built up in the usual way from the above operators and connectives, according to the following grammar:

$$\varphi := p \mid \neg \varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \mathbf{X} \varphi \mid \varphi \mathbf{U} \varphi \mid \varphi \mathbf{R} \varphi,$$

where  $p$  is an atomic proposition in AP. By  $|\varphi|$  we denote the length of  $\varphi$ .

<sup>3</sup><http://wpage.unina.it/m.faella/nycs>.

The semantics of LTL is typically given *w.r.t.* infinite sequences (*i.e.*, words) of sets of atomic propositions in AP, *a.k.a.* *discrete traces*, to capture properties of discrete infinite computations. Since we are interested in the verification of continuous systems, we shall also consider a semantics based on signals, in a similar vein to some previous works [4]. In Section 8, we describe how the discrete and the continuous semantics are related, a connection that we leverage to reduce verification of continuous properties to a combination of verification of discrete properties and geometric reasoning. Besides the classic semantics on infinite-time signals, for both the discrete and the continuous versions, we shall also consider the finite-time semantic fragments, where formulae are interpreted over finite traces and finite-time signals, respectively.

We shall provide below a unified semantics for the language that encompasses all these variants (discrete, continuous, finite-time and infinite-time). This is done by introducing a generalised notion of signal, called *abstract signal*, which assigns observations (*i.e.*, a set of atomic propositions) to each element of some interval of totally ordered points and can accommodate both discrete traces and continuous signals. To reflect this generality, we shall refer to the resulting framework as *Abstract Linear Time Logic* (ALTL, for short).

*Abstract Semantics.* As time domain of ALTL, we consider any total order  $(\mathcal{I}, \leq)$  with a minimum element 0 and maximum element  $\infty$ . Concrete examples of time domains are  $(\mathbb{R}_{\geq 0} \cup \{\infty\}, \leq)$  and  $(\mathbb{N} \cup \{\infty\}, \leq)$ .

An *interval* of  $\mathcal{I}$  is a subset  $I \subseteq \mathcal{I} \setminus \{\infty\}$  with the following property: there exist two elements  $a, b \in \mathcal{I}$  such that:

- (i)  $d \in I$ , for all  $d \in \mathcal{I}$  such that  $a < d < b$ , and
- (ii)  $a \leq d \leq b$ , for all  $d \in I$ .

We may write  $\langle a, b \rangle$  to denote any interval  $I$  such that  $\inf(I) = a$  and  $\sup(I) = b$ . We call an interval  $I$  *unbounded* if  $\sup(I) = \infty$ , and *bounded* otherwise. Note that,  $a$  (*resp.*,  $b$ ) may not belong to  $I$ , in which case we say that  $I$  is a left-open (*resp.*, right-open) interval.

The semantics of ALTL formulae is given with respect to an abstract signal  $\sigma: I \rightarrow 2^{\text{AP}}$ , where  $I$  is an interval of  $\mathcal{I}$ . A signal is *left-open* if so is its domain interval  $I$ , namely if  $\inf(I) \notin I$ , otherwise it is *left-closed*. The semantic conditions are defined as follows:

- $\sigma \models p$ , for  $p \in \text{AP}$ , if and only if:
  - +  $p \in \sigma(\inf(I))$ , if  $\inf(I) \in I$ ;
  - + there exists  $d \in I$  such that  $p \in \sigma(d')$ , for all  $d' \in I$  with  $d' \leq d$ , if  $\inf(I) \notin I$ ;
- $\sigma \models \mathbf{X}\varphi$  if and only if there exists  $d \in I \setminus \{\inf(I)\}$  such that  $\sigma_{\geq d'} \models \varphi$ , for all  $d' \in I \setminus \{\inf(I)\}$  with  $d' \leq d$ ;
- $\sigma \models \varphi_1 \mathbf{U} \varphi_2$  if and only if there exists  $d \in I$  such that  $\sigma_{\geq d} \models \varphi_2$  and  $\sigma_{\geq d'} \models \varphi_1$ , for all  $d' \in I$  with  $d' < d$ ;

- $\sigma \models \varphi_1 \mathbf{R} \varphi_2$  if and only if for all  $d \in I$ , it holds  $\sigma_{\geq d} \models \varphi_2$  or there exists  $d' \in I$  with  $d' < d$  such that  $\sigma_{\geq d'} \models \varphi_1$ .

The base case for left-closed signals (*i.e.*, when  $\inf(I) \in I$ ) is standard. For a left-open signal, instead, we stipulate that atomic proposition  $p \in \text{AP}$  is satisfied if there exists an initial left-open subinterval contained in the domain  $I$  of the signal where  $p$  is observed. The case for the next operator coincides with the classic LTL semantics of  $\mathbf{X}$  when the domain  $I$  is a discrete linear order. For non-discrete signals, the operator essentially predicates about the left-open interval adjacent to the current point in every point of which the argument is required to be satisfied. The intuition is that the argument must hold immediately after  $\inf(I)$ , *i.e.*, the initial point of the interval  $I$ . Clearly, if the abstract signal is bounded, in the last point of the signal no next formula can be satisfied. Observe that, on left-open signals, the semantics of  $\varphi$  and  $\mathbf{X}\varphi$  always coincide.

We shall also consider the *non-recurrent* fragment of ALTL, which contains the positive normal form formulae, where in each release subformula of the form  $\varphi_1 \mathbf{R} \varphi_2$ , the second argument  $\varphi_2$  is required to be a propositional formula. This fragment contains ALTL formulae that disallow temporal properties to be required continuously, except for plain boolean ones. That means, for instance, that the formula  $\perp \mathbf{R} (p \wedge q)$ , which is equivalent to  $\mathbf{G} (p \wedge q)$ , belongs to the fragment, whereas  $\perp \mathbf{R} (\mathbf{F} p)$ , which is equivalent to  $\mathbf{G} \mathbf{F} p$ , does not since the truth of the temporal formula  $\mathbf{F} p$  is required continuously. The fragment we are considering is an extension of the flat fragment of LTL introduced in [26], where also the first argument of a  $\mathbf{U}$ -formula is required to be propositional.

*Concrete Semantics.* The abstract semantics just described can be instantiated, by fixing the time domain and the set of allowed intervals, so as to capture various concrete semantics for the language. In particular, we obtain the following concrete semantics:

$\text{LTL}_f$ : time domain  $(\mathbb{N} \cup \{\infty\}, \leq)$ , bounded intervals;

$\text{LTL}_\omega$ : time domain  $(\mathbb{N} \cup \{\infty\}, \leq)$ , unbounded intervals;

$\text{LTL}_\infty$ : time domain  $(\mathbb{N} \cup \{\infty\}, \leq)$ , all intervals;

$\text{RTL}_f$ : time domain  $(\mathbb{R}_{\geq 0} \cup \{\infty\}, \leq)$ , bounded intervals;

$\text{RTL}_\omega$ : time domain  $(\mathbb{R}_{\geq 0} \cup \{\infty\}, \leq)$ , unbounded intervals;

$\text{RTL}_\infty$ : time domain  $(\mathbb{R}_{\geq 0} \cup \{\infty\}, \leq)$ , all intervals.

The logic  $\text{LTL}_f$  denotes the logic interpreted over finite traces [6], whereas  $\text{LTL}_\omega$  corresponds to standard LTL interpreted on infinite traces.  $\text{LTL}_\infty$ , instead, interprets formulae over both finite and infinite traces. Similarly,  $\text{RTL}_f$  interprets formulae over finite-time continuous signals,  $\text{RTL}_\omega$  over infinite-time continuous signals and  $\text{RTL}_\infty$  over arbitrary (*i.e.*, both finite-time and infinite-time) continuous signals. The semantics of  $\text{RTL}_f$  essentially corresponds to the logic by the same name from [4], except that we consider both left-open and left-closed signals and we omit the past operator *Since*.

**Remark 1.** *Contrary to [4], our version of RTL includes a primitive next operator  $X$  instead of the strict until operator  $\dot{U}$  employed there and whose semantics is the following:*

$$\begin{aligned} \sigma \models \varphi_1 \dot{U} \varphi_2 \quad \text{iff} \quad & \text{there exists } d \in I \setminus \text{inf}(I) \text{ such that } \sigma_{\geq d} \models \varphi_2 \\ & \text{and } \sigma_{\geq d'} \models \varphi_1, \text{ for all } d' \in I \setminus \text{inf}(I) \text{ with } d' < d \end{aligned} \quad (2)$$

In classic LTL, the operator  $\dot{U}$  is inter-derivable with the non-strict until operator  $U$ , thanks to the next operator. Indeed,  $\varphi_1 U \varphi_2 \equiv \varphi_2 \vee (\varphi_1 \wedge \varphi_1 \dot{U} \varphi_2)$  and  $\varphi_1 \dot{U} \varphi_2 \equiv X(\varphi_1 U \varphi_2)$ . However, in the version of RTL considered in [4] the second equivalence cannot be stated, since the operator  $X$  is not available. It is easy to check that the interpretation of the ALTL formula  $X(\varphi_1 U \varphi_2)$  on continuous signals is indeed equivalent to the interpretation of  $\varphi_1 \dot{U} \varphi_2$ .

**Theorem 1** ([5, 27]). *For all  $\text{LTL}_\omega$  (resp.,  $\text{LTL}_f$ ) formulae  $\varphi$  there exists a Büchi (resp., a finite) automaton  $\mathcal{A}_\varphi$  that accepts all and only the models of  $\varphi$ .*

### 3.1. The Model-Checking Problem

We are interested in solving the model-checking problem for an RTL formula  $\varphi$  against a polyhedral system and under one of the semantics  $\gamma \in \{\text{fin}, \text{inf}, \text{may}, \text{must}\}$ , as described in the previous section. For a point  $x \in \text{Inv} \subseteq \mathbb{R}^n$ , the problem requires to decide whether  $x \models^\gamma \varphi$ , namely if there exists a trajectory in  $\text{Traj}^\gamma(x)$  from that point whose induced signal satisfies the formula. Formally, the relation  $\models^\gamma$  is defined as follows:

$$x \models^\gamma \varphi \text{ if and only if } \exists f \in \text{Traj}^\gamma(x). \text{ such that } \sigma_f \models \varphi$$

More specifically, we shall solve the more general problem of computing the *existential denotation* of an RTL formula *w.r.t.* a polyhedral system  $\mathcal{P}$  and a semantics  $\gamma$  defined as follows.

**Definition 1.** *Given an RTL formula  $\varphi$ , a polyhedral system  $\mathcal{P}$  on the same set of atomic propositions, and a semantics  $\gamma \in \{\text{fin}, \text{inf}, \text{may}, \text{must}\}$ , the existential denotation  $\llbracket \varphi \rrbracket_\exists^\gamma$  of  $\varphi$  on  $\mathcal{P}$  w.r.t. the semantics  $\gamma$  is the set of points  $x \in \text{Inv} \subseteq \mathbb{R}^n$  such that  $x \models^\gamma \varphi$ .*

Clearly, the universal denotation  $\llbracket \varphi \rrbracket_\forall^\gamma$  of a formula, namely the set of points  $x$  from which (the signals of) all trajectories in  $\text{Traj}^\gamma(x)$  satisfy  $\varphi$ , can easily be computed by duality, that is  $\llbracket \varphi \rrbracket_\forall^\gamma = \text{Inv} \setminus \llbracket \neg \varphi \rrbracket_\exists^\gamma$ .

In this paper we show that the problem is decidable for each semantics, possibly under some additional assumptions on the underlying polyhedral system  $\mathcal{P}$  and/or the class of formulae considered. In particular, decidability is shown in the following cases:

1. The semantics  $\gamma = \text{fin}$  without any additional assumptions on  $\mathcal{P}$  or on the formulae.
2. The semantics  $\gamma = \text{inf}$  under the following assumptions:

- (a) **Omnidirectional Flow**, that is the interior of the flow of  $\mathcal{P}$  contains the origin, and no restriction on the formulae. In this scenario the system is allowed to stop at any moment and remain still and to move in any direction;
  - (b) **Non-Recurrent RTL and Closed Flow**, where we only consider formulae of the non-recurrent fragment of RTL and the flow of  $\mathcal{P}$  is a closed convex polyhedron.
3. The semantics  $\gamma \in \{\text{may}, \text{must}\}$  under the following assumptions:
- (a) **Omnidirectional Flow** and no restriction on the formulae;
  - (b) **Forced Motion and Bounded Invariant**, namely the closure of the flow of  $\mathcal{P}$  does not contain the origin (*i.e.*,  $\mathbf{0} \notin \text{cl}(\text{Flow})$ ) and the invariant is bounded with no restriction on the formulae. In this scenario the system cannot move arbitrarily slow, nor it can stop as long as it stays inside the invariant. The additional restriction on the invariant intuitively ensures that any trajectory will eventually reach the edge of the invariant and stop there, hence only finite-time trajectories exist;
  - (c) **Non-Recurrent RTL and Closed Flow**.

The model-checking problem remains an **open problem** for the semantics  $\{\text{inf}, \text{may}, \text{must}\}$  and no restrictions on the polyhedral system or on the formula.

#### 4. Discretisation

To address the model-checking problem of a polyhedral system against an RTL specification, we reduce it to a suitable decision problem for the discrete versions of the logic, *i.e.*, classic  $\omega$ -regular LTL and its finite variant  $\text{LTL}_f$ . Specifically, we show that, for all RTL formulae  $\varphi$  on a set of atomic propositions  $AP$ , there exists an LTL formula  $\text{dsc}(\varphi)$  on the extended set  $AP \cup \{\text{sing}\}$  such that a signal  $\sigma$  satisfies  $\varphi$  *iff* the discrete traces induced by  $\sigma$  satisfy  $\text{dsc}(\varphi)$ . This result is proved at the end of this section as Theorem 2. In order to do this, we first need to define and characterise the discrete versions of signals (Section 4.1) and formulae (Section 4.2).

##### 4.1. Discretising Signals

Recall from Section 2.1 that a time slicing  $\tau = \{t_i\}_{i=0} \in \text{TS}(\sigma)$  of a signal  $\sigma: I \rightarrow 2^{AP}$  decomposes  $\sigma$  into a sequence of slices corresponding to an alternation of singular and open time intervals. Such a sequence is finite, if  $\sigma$  is a finite-time signal, and infinite, otherwise. Now, let us introduce the function  $\text{slice}_\sigma^\tau: I \rightarrow \mathbb{N}$ , unambiguously associating each time instant  $t \in I$  with the index of its slice  $\text{slice}_\sigma^\tau(t)$ . Formally:

$$\text{slice}_\sigma^\tau(t) = \begin{cases} 2i, & \text{if } t = t_i; \\ 2i + 1, & \text{if } t \in (t_i, t_{i+1}). \end{cases}$$

We can define the discrete trace  $trc(\sigma, \tau)$ , and consequently  $trc(f, \tau) \triangleq trc(\sigma_f, \tau)$ , by lumping together in a single object the time instants of each open interval  $(t_i, t_{i+1})$  and inserting between any two such intervals the observables of the singular time point separating them. We maintain the distinction between open and singular intervals by means of an accessory atomic proposition *sing* that holds true in all and only the time points  $t_i$  of the time slicing  $\tau$ . Denote again by  $\alpha_i$  the set of observables true in the open interval  $(t_i, t_{i+1})$ . The discretisation  $trc(\sigma, \tau)$ , defined below for both left-closed and left-open signals, is a finite or infinite word depending on whether  $\sigma$  is a finite-time or infinite-time signal. We use  $trc(\sigma, \tau)_i \subseteq AP \cup \{sing\}$  to denote the  $i$ -th symbol of the discrete trace. Formally, for a left-closed signal  $\sigma: I \rightarrow 2^{AP}$  and an index  $j \in \text{rng}(slice_\sigma^\tau)$ , we set:

$$trc(\sigma, \tau)_j \triangleq \begin{cases} \sigma(t_i) \cup \{sing\}, & \text{if } j \text{ is even and } i = j/2; \\ \alpha_i, & \text{if } j \text{ is odd and } i = (j-1)/2. \end{cases}$$

For a left-open signal  $\sigma: I \rightarrow 2^{AP}$  and an index  $j \in \text{rng}(slice_\sigma^\tau)$ , we set:

$$trc(\sigma, \tau)_j \triangleq \begin{cases} \alpha_i, & \text{if } j \text{ is even and } i = j/2; \\ \sigma(t_i) \cup \{sing\}, & \text{if } j \text{ is odd } i = (j+1)/2. \end{cases}$$

Consider the finite-time left-closed signal  $\sigma: [0, 18] \rightarrow 2^{AP}$  depicted in Figure 3 taking values in  $\{\emptyset, \{t_{10}\}, \{p_1\}, \{p_2\}, \{t_0, t_{10}\}, \{p_1, t_{10}\}, \{p_2, t_{10}\}\} \subseteq 2^{AP}$ , where  $AP = \{t_0, t_{10}, p_1, p_2\}$ , and the corresponding minimal time-slicing  $\tau_1 = \{0, 1, 7, 8, 10, 11, 12.5, 14, 15, 17, 18\} \in TS(\sigma)$  already discussed on Page 12. At this point, by applying the definition just introduced, it is immediate to see that the corresponding trace is:

$$\begin{aligned} trc(\sigma, \tau_1) = & \{sing, t_0, t_{10}\}, \{t_{10}\}, \{sing, p_1, t_{10}\}, \{p_1, t_{10}\}, \{sing, p_1, t_{10}\}, \{t_{10}\}, \\ & \{sing, p_2, t_{10}\}, \{p_2, t_{10}\}, \{sing, p_2, t_{10}\}, \emptyset, \{sing, p_1\}, \emptyset, \{sing, p_2\}, \{p_2\}, \\ & \{sing, p_2\}, \emptyset, \{sing, p_1\}, \emptyset, \{sing, p_2\}, \emptyset, \{sing\}. \end{aligned}$$

Before continuing with the discretisation of the specification, we state a commutativity property enjoyed by the composition of the discretisation function with the suffix operation on signals, time slicings, and traces. In particular, for some  $t \in I$ , we define  $(\{t_i\}_{i=0})_{\geq t} \triangleq \{t'_i\}_{i=0}$ , with  $t'_0 \triangleq t$ , and  $t'_i \triangleq t_{i+l}$ , where  $l \in \mathbb{N}$  is the maximum index such that  $t_l \leq t$ , which also corresponds to the value  $\left\lfloor \frac{slice_\sigma^\tau(t)}{2} \right\rfloor$ . Note that  $\tau_{\geq t} \in TS(\sigma_{\geq t}) = TS(\sigma_{> t})$ , given  $\tau \in TS(\sigma)$ .

**Lemma 4.** *Let  $\sigma: I \rightarrow 2^{AP}$  be a signal,  $\tau \in TS(\sigma)$  one of its time slicings,  $t \in I$  a time instant in the signal domain, and  $h = slice_\sigma^\tau(t)$  the corresponding slice index. Then, it holds true that:*

$$trc(\sigma_{> t}, \tau_{\geq t}) = \begin{cases} trc(\sigma, \tau)_{\geq h+1}, & \text{if } sing \in trc(\sigma, \tau)_h; \\ trc(\sigma, \tau)_{\geq h}, & \text{otherwise.} \end{cases}$$

The following corollary is an immediate consequence of the above lemma and the two equalities  $\text{trc}(\sigma_{\geq t}, \tau_{\geq t}) = (\sigma(t) \cup \{\text{sing}\}) \cdot \text{trc}(\sigma_{> t}, \tau_{\geq t})$  and  $\text{trc}(\sigma, \tau)_{\geq h} = (\sigma(t) \cup \{\text{sing}\}) \cdot \text{trc}(\sigma, \tau)_{\geq h+1}$ , whenever  $h = \text{slice}_{\sigma}^{\tau}(t)$  and  $\text{sing} \in \text{trc}(\sigma, \tau)_h$ .

**Corollary 1.** *Let  $\sigma: I \rightarrow 2^{AP}$  be a signal,  $\tau \in TS(\sigma)$  one of its time slicings,  $t \in I$  a time instant in the signal domain, and  $h = \text{slice}_{\sigma}^{\tau}(t)$  the corresponding slice index. Then, it holds true that:*

$$\text{trc}(\sigma, \tau)_{\geq h} = \begin{cases} \text{trc}(\sigma_{\geq t}, \tau_{\geq t}), & \text{if } \text{sing} \in \text{trc}(\sigma, \tau)_h; \\ \text{trc}(\sigma_{> t}, \tau_{\geq t}), & \text{otherwise.} \end{cases}$$

In addition, it is immediate to see that, due to its specific definition, a trace of a signal satisfies the following property concerning the auxiliary *sing* atomic proposition: (a) singular and open intervals alternate throughout the trace; (b) the trace starts in a singular interval *iff* the underlying signal is left-closed; (c) the trace ends in a singular interval *iff* the underlying signal is right-closed.

**Proposition 5.** *For a signal  $\sigma: I \rightarrow 2^{AP}$  and a time slicing  $\tau \in TS(\sigma)$ , it holds that  $\text{trc}(\sigma, \tau) \models \mathbf{G}((\text{sing} \leftrightarrow \mathbf{X}\neg\text{sing}) \vee \text{last})$ , where  $\text{last} \triangleq \neg\mathbf{X}\top$ . Moreover,  $\sigma$  is left-closed (resp., right-closed) *iff*  $\text{trc}(\sigma, \tau) \models \text{sing}$  (resp.,  $\text{trc}(\sigma, \tau) \models \mathbf{F}(\text{last} \wedge \text{sing})$ ).*

#### 4.2. Discretising Formulae

We can now introduce the required transformation from RTL to LTL. Intuitively, this translation exploits the segmentation induced by a time slicing of a signal to verify whether the changes of observables along the signal actually satisfy the property prescribed by the RTL formula. Formally, we set the following:

$$\begin{aligned} \mathbf{dsc}(p) &\triangleq p; \\ \mathbf{dsc}(\neg\varphi) &\triangleq \neg\mathbf{dsc}(\varphi); \\ \mathbf{dsc}(\varphi_1 \wedge \varphi_2) &\triangleq \mathbf{dsc}(\varphi_1) \wedge \mathbf{dsc}(\varphi_2); \\ \mathbf{dsc}(\varphi_1 \vee \varphi_2) &\triangleq \mathbf{dsc}(\varphi_1) \vee \mathbf{dsc}(\varphi_2); \\ \mathbf{dsc}(\mathbf{X}\varphi) &\triangleq (\text{sing} \wedge \mathbf{X}\mathbf{dsc}(\varphi)) \vee (\neg\text{sing} \wedge \mathbf{dsc}(\varphi)); \\ \mathbf{dsc}(\varphi_1 \mathbf{U} \varphi_2) &\triangleq \mathbf{dsc}(\varphi_1) \mathbf{U} ((\text{sing} \vee \mathbf{dsc}(\varphi_1)) \wedge \mathbf{dsc}(\varphi_2)); \\ \mathbf{dsc}(\varphi_1 \mathbf{R} \varphi_2) &\triangleq \mathbf{dsc}(\varphi_1) \mathbf{R} ((\neg\text{sing} \wedge \mathbf{dsc}(\varphi_1)) \vee \mathbf{dsc}(\varphi_2)). \end{aligned}$$

As an illustration of the transformation, consider the RTL formula  $\mathbf{X}(p \mathbf{U} q)$ . By a straightforward application of the fifth and sixth clauses of the above definition, one can easily verify that  $\mathbf{dsc}(\mathbf{X}(p \mathbf{U} q))$  is equal to

$$(\text{sing} \wedge \mathbf{X}(p \mathbf{U} ((\text{sing} \vee p) \wedge q))) \vee (\neg\text{sing} \wedge (p \mathbf{U} ((\text{sing} \vee p) \wedge q))).$$

At this point, a remark is in order. The transformation defined above may, in principle, lead to an exponential blowup in the length of the resulting formula. This is due to the repeated occurrences of the subformula  $\mathbf{dsc}(\varphi)$  in

the translation rule for the  $\mathbf{X}$  operator, as well as of  $\mathbf{dsc}(\varphi_1)$  in the rules for the  $\mathbf{U}$  and  $\mathbf{R}$  operators. Nevertheless, the number of distinct subformulae occurring in  $\mathbf{dsc}(\varphi)$ , *i.e.*, what is known as *DAG-size*, remains linear in the DAG-size of the original formula  $\varphi$ . This is a crucial property, since the DAG-size is typically regarded as the relevant complexity parameter in automata-theoretic constructions for temporal logics.

To prove the correctness of the above transformation, we first need to state three properties enjoyed by the semantics of RTL. In the following, we say that a signal  $\sigma: I \rightarrow 2^{AP}$  is *B-uniform*, for an interval  $B \subseteq I$ , if  $\sigma(t) = \sigma(t')$ , for all  $t, t' \in B$ .

The first lemma states that RTL cannot distinguish between two suffixes of a signal whose starting time instants are contained in the same uniform interval.

**Lemma 5.** *For all RTL formulae  $\varphi$ , signals  $\sigma: I \rightarrow 2^{AP}$ , and open intervals  $B \subseteq I$  such that  $\sigma$  is B-uniform, the following holds true:  $\sigma_{\sim_1 t_1} \models \varphi$  iff  $\sigma_{\sim_2 t_2} \models \varphi$ , for all  $t_1, t_2 \in B$  and  $\sim_1, \sim_2 \in \{\geq, >\}$ .*

The next lemma connects satisfaction of an RTL formula *w.r.t.* a left-open signal with satisfaction *w.r.t.* left-closed suffixes of that signal.

**Lemma 6.** *For all RTL formulae  $\varphi$ , signals  $\sigma: I \rightarrow 2^{AP}$ , and time instants  $t \in I$ , the following holds true:  $\sigma_{>t} \models \varphi$  iff there exists a time instant  $t' \in I$ , with  $t' > t$ , such that  $\sigma_{\geq t'} \models \varphi$ , for all  $t'' \in (t, t']$ .*

The last lemma reformulates equivalently the RTL semantics of the next operator, as given in Section 3, in terms of left-open signals.

**Lemma 7.** *For all RTL formulae  $\varphi$  and signals  $\sigma: I \rightarrow 2^{AP}$ , the following holds true:  $\sigma \models \mathbf{X}\varphi$  iff  $\sigma_{>\inf(I)} \models \varphi$ .*

By leveraging the above three lemmata, we can establish the correctness of the discretisation. This allows us to reduce verification of RTL properties against signals to verification of LTL properties against discrete traces, as described in the next sections.

**Theorem 2.** *For all RTL formulae  $\varphi$ , signals  $\sigma$ , and time slicings  $\tau \in TS(\sigma)$ , it holds that  $\sigma \models \varphi$  iff  $\text{trc}(\sigma, \tau) \models \mathbf{dsc}(\varphi)$ .*

*Proof.* The proof proceeds by structural induction on the formula, where we consider an arbitrary (finite or infinite) time slicing  $\tau = \{t_i\}_{i \geq 0}$  of  $\sigma: I \rightarrow 2^{AP}$ , with  $a \triangleq \inf(I)$ . Since the inductive cases of the Boolean operators  $\neg$ ,  $\wedge$ , and  $\vee$  are trivial to deal with and thanks to the duality property between  $\mathbf{U}$  and  $\mathbf{R}$ , we focus on the atomic propositions and the two temporal operators  $\mathbf{X}$  and  $\mathbf{U}$ .

- **[Base case  $\varphi = p \in AP$ ].** We distinguish the two cases of left-closed and left-open signals.
  - + **[ $\sigma$  is left-closed].** By definition of  $\text{trc}(\sigma, \tau)$ , it holds that  $\text{trc}(\sigma, \tau)_0 = \sigma(t_0) \cup \{\text{sing}\}$ . Hence, being  $\mathbf{dsc}(p) = p$  and  $t_0 = a$ , we have that  $\sigma \models p$  iff  $p \in \sigma(a)$  iff  $\text{trc}(\sigma, \tau) \models \mathbf{dsc}(p)$ .

+ [ **$\sigma$  is left-open**]. Observe that  $\text{trc}(\sigma, \tau)_0 = \sigma(t)$ , for every  $t \in (t_0, t_1)$ , since, by definition of time slicing of  $\sigma$ , the observables are constant in each open interval  $(t_i, t_{i+1})$ . Now,  $\sigma \models p$  iff  $p \in \sigma(t)$ , for all  $t \in (a, t) = (t_0, t_1)$ , for some  $t > a$ . Then, it immediately follows that  $\sigma \models p$  iff  $\text{trc}(\sigma, \tau) \models \mathbf{dsc}(p)$ .

- [**Inductive case  $\varphi = \mathbf{X}\varphi'$** ]. By Lemma 7, it holds that  $\sigma \models \varphi$  iff  $\sigma_{>a} \models \varphi'$ , since  $a = \inf(I)$ . Recall that  $\tau_{\geq t} \in TS(\sigma_{\geq t}) = TS(\sigma_{>t})$ , for all  $t \in I$ . By the inductive hypothesis, we have then that  $\sigma_{>a} \models \varphi'$  iff  $\text{trc}(\sigma_{>a}, \tau_{\geq a}) \models \mathbf{dsc}(\varphi')$ , so,  $\sigma \models \varphi$  iff  $\text{trc}(\sigma_{>a}, \tau_{\geq a}) \models \mathbf{dsc}(\varphi')$ . We now proceed by a case analysis on the left-openness of  $\sigma$ .

+ [ **$\sigma$  is left-closed**]. Obviously,  $\text{sing} \in \text{trc}(\sigma, \tau)_0$ , thus,  $\text{trc}(\sigma, \tau) \models \text{sing}$ . Moreover, by Lemma 4, we have that  $\text{trc}(\sigma_{>a}, \tau_{\geq a}) = \text{trc}(\sigma, \tau)_{\geq 1}$ , since  $\text{slice}_\sigma^\tau(a) = 0$ , which implies that  $\sigma \models \varphi$  iff  $\text{trc}(\sigma, \tau)_{\geq 1} \models \mathbf{dsc}(\varphi')$  and, so,  $\sigma \models \varphi$  iff  $\text{trc}(\sigma, \tau) \models \mathbf{X}\mathbf{dsc}(\varphi')$ . Therefore, we have  $\sigma \models \varphi$  iff  $\text{trc}(\sigma, \tau) \models \text{sing} \wedge \mathbf{X}\mathbf{dsc}(\varphi')$ .

+ [ **$\sigma$  is left-open**]. Obviously,  $\sigma_{>a} = \sigma$ , hence,  $\sigma \models \varphi$  iff  $\text{trc}(\sigma, \tau) \models \mathbf{dsc}(\varphi')$ . Moreover,  $\text{sing} \notin \text{trc}(\sigma, \tau)_0$ , which means that  $\text{trc}(\sigma, \tau) \models \neg \text{sing}$ . As a consequence, we obtain  $\sigma \models \varphi$  iff  $\text{trc}(\sigma, \tau) \models \neg \text{sing} \wedge \mathbf{dsc}(\varphi')$ .

Summing up, we have that  $\sigma \models \varphi$  iff  $\text{trc}(\sigma, \tau) \models (\text{sing} \wedge \mathbf{X}\mathbf{dsc}(\varphi')) \wedge (\neg \text{sing} \wedge \mathbf{dsc}(\varphi'))$  iff  $\text{trc}(\sigma, \tau) \models \mathbf{dsc}(\varphi)$  as required by the statement.

- [**Inductive case  $\varphi = \varphi_1 \mathbf{U} \varphi_2$** ]. For this final inductive case, for the sake of simplicity, we split the proof in its two directions.

+ [**Only if direction**]. Since  $\sigma \models \varphi$ , by the semantics of the temporal operator  $\mathbf{U}$ , there exists  $t \in I$  such that  $\sigma_{\geq t} \models \varphi_2$  and, for all  $t' \in I$ , with  $t' < t$ , it holds  $\sigma_{\geq t'} \models \varphi_1$ . We now need to split the initial part of the proof in two further subcases, depending on whether  $t$  belongs to the time slicing  $\tau$  or is contained in one of its open intervals.

\* [ $t \in \tau$ ]. Let  $j$  be the position in the discrete trace corresponding to the instant  $t$ , i.e.,  $j \triangleq \text{slice}_\sigma^\tau(t)$ . Obviously,  $\text{sing} \in \text{trc}(\sigma, \tau)_j$ . By the inductive hypothesis,  $\text{trc}(\sigma_{\geq t}, \tau_{\geq t}) \models \mathbf{dsc}(\varphi_2)$ . Moreover,  $\text{trc}(\sigma_{\geq t}, \tau_{\geq t}) = \text{trc}(\sigma, \tau)_{\geq j}$ , by Corollary 1. Hence,  $\text{trc}(\sigma, \tau)_{\geq j} \models \text{sing} \wedge \mathbf{dsc}(\varphi_2)$ .

\* [ $t \notin \tau$ ]. If, on the other hand,  $t \in (t_i, t_{i+1})$ , for some index  $i$ , then  $\sigma$  is clearly  $\mathbf{B}$ -uniform, if we take  $\mathbf{B} = (t_i, t] \subset (t_i, t_{i+1})$ . Hence, by Lemma 5, we have  $\sigma_{\geq t} \models \varphi_2$ , for all  $t \in \mathbf{B}$  and, by Lemma 6 and the fact that  $\inf(\mathbf{B}) = t_i$ , we conclude  $\sigma_{>t_i} \models \varphi_2$ . Taking  $j \triangleq \text{slice}_\sigma^\tau(t) = \text{slice}_\sigma^\tau(t_i) + 1$ , we have that  $\text{sing} \notin \text{trc}(\sigma, \tau)_j$ . By applying the inductive hypothesis, we obtain  $\text{trc}(\sigma_{>t_i}, \tau_{\geq t_i}) \models \mathbf{dsc}(\varphi_2)$ . In this case, we know from the assumption that  $\sigma_{\geq t'} \models \varphi_1$ , for all  $t_i < t' < t$ . Then, by applying again Lemmata 5 and 6, we obtain that  $\text{trc}(\sigma_{>t_i}, \tau_{\geq t_i}) \models \mathbf{dsc}(\varphi_1)$ . Again by Corollary 1,  $\text{trc}(\sigma_{>t_i}, \tau_{\geq t_i}) = \text{trc}(\sigma, \tau)_{\geq j}$ . Thus, we can conclude  $\text{trc}(\sigma, \tau)_{\geq j} \models \mathbf{dsc}(\varphi_1) \wedge \mathbf{dsc}(\varphi_2)$ .

Regardless of the case, we have obtained that  $\text{trc}(\sigma, \tau)_{\geq \bar{j}} \models (\text{sing} \vee \mathbf{dsc}(\varphi_1)) \wedge \mathbf{dsc}(\varphi_2)$ , where  $\bar{j} \triangleq \text{slice}_\sigma^\tau(t)$ . At this point, we split the final part of the proof in two subcases, depending on whether  $\sigma$  is left-closed.

- \* **[ $\sigma$  is left-closed].** Let us now consider any  $t' \in J$ , where  $J = [t_0, t_i)$ , if  $t = t_i$ , and  $J = [t_0, t_i]$ , if  $t \in (t_i, t_{i+1})$ , for some index  $i$ . We have two cases, depending on whether  $t' = t_j$  or  $t' \in (t_j, t_{j+1})$ , for some  $0 \leq j < i$ . By applying to  $t'$  the same reasoning we applied to  $t$  above, using the inductive hypothesis, Corollary 1 and Lemmata 5 and 6, we obtain that  $\text{trc}(\sigma, \tau)_{\geq j} \models \mathbf{dsc}(\varphi_1)$ , for  $j = \text{slice}_\sigma^\tau(t')$ . Since, in addition,  $\{\text{slice}_\sigma^\tau(t') : t' \in J\} = \{0, \dots, \bar{j} - 1\}$ , we can conclude that  $\text{trc}(\sigma, \tau)_{\geq j} \models \mathbf{dsc}(\varphi_1)$ , for all  $0 \leq j < \bar{j}$ , thanks to the definition of  $\text{trc}(\sigma, \tau)$  for left-closed signals  $\sigma$ .
- \* **[ $\sigma$  is left-open].** For this case, the proof proceeds exactly as the previous one, where we consider the sets  $J = (t_0, t_i)$ , if  $t = t_i$ , and  $J = [t_0, t_i]$ , if  $t \in (t_i, t_{i+1})$ , for some index  $i$ , deriving  $\{\text{slice}_\sigma^\tau(t') : t' \in J\} = \{1, \dots, \bar{j} - 1\}$ . Thanks to the definition of  $\text{trc}(\sigma, \tau)$  for left-open signals  $\sigma$ , we then obtain  $\text{trc}(\sigma, \tau)_{\geq j} \models \mathbf{dsc}(\varphi_1)$ , for all  $0 \leq j < \bar{j}$ .

Regardless of the case,  $\text{trc}(\sigma, \tau)_{\geq j} \models \mathbf{dsc}(\varphi_1)$ , for all  $0 \leq j < \bar{j}$ . Together with  $\text{trc}(\sigma, \tau)_{\geq \bar{j}} \models (\text{sing} \vee \mathbf{dsc}(\varphi_1)) \wedge \mathbf{dsc}(\varphi_2)$ , this gives us  $\text{trc}(\sigma, \tau) \models \mathbf{dsc}(\varphi_1) \cup ((\text{sing} \vee \mathbf{dsc}(\varphi_1)) \wedge \mathbf{dsc}(\varphi_2))$  and, finally,  $\text{trc}(\sigma, \tau) \models \mathbf{dsc}(\varphi_1 \cup \varphi_2)$ .

+ **[If direction].** Assume  $\text{trc}(\sigma, \tau) \models \mathbf{dsc}(\varphi_1 \cup \varphi_2)$ , i.e.,  $\text{trc}(\sigma, \tau) \models \mathbf{dsc}(\varphi_1) \cup ((\text{sing} \vee \mathbf{dsc}(\varphi_1)) \wedge \mathbf{dsc}(\varphi_2))$ . By the semantics of the temporal operator  $\cup$ , there exists an index  $j$  such that  $\text{trc}(\sigma, \tau)_{\geq j} \models (\text{sing} \vee \mathbf{dsc}(\varphi_1)) \wedge \mathbf{dsc}(\varphi_2)$  and  $\text{trc}(\sigma, \tau)_{\geq z} \models \mathbf{dsc}(\varphi_1)$ , for all  $z < j$ . As for the previous direction, we have two further subcases, depending on whether  $\text{trc}(\sigma, \tau)_j$  contains the proposition *sing* or not.

- \* **[ $\text{sing} \in \text{trc}(\sigma, \tau)_j$ ].** Obviously,  $\text{trc}(\sigma, \tau)_{\geq j} \models \text{sing} \wedge \mathbf{dsc}(\varphi_2)$  and  $j = \text{slice}_\sigma^\tau(t_i)$ , for some  $i$ . By Corollary 1,  $\text{trc}(\sigma, \tau)_{\geq j} = \text{trc}(\sigma_{\geq t_i}, \tau_{\geq t_i})$ . Therefore,  $\text{trc}(\sigma_{\geq t_i}, \tau_{\geq t_i}) \models \mathbf{dsc}(\varphi_2)$  and, then,  $\sigma_{\geq t_i} \models \varphi_2$ , thanks to the inductive hypothesis.
- \* **[ $\text{sing} \notin \text{trc}(\sigma, \tau)_j$ ].** Obviously,  $\text{trc}(\sigma, \tau)_{\geq j} \models \mathbf{dsc}(\varphi_1) \wedge \mathbf{dsc}(\varphi_2)$  and  $j = \text{slice}_\sigma^\tau(t)$ , for all  $t \in (t_i, t_{i+1})$  and some  $i$ . For all such  $t$ , then, we obtain  $\text{trc}(\sigma, \tau)_{\geq j} = \text{trc}(\sigma_{> t}, \tau_{\geq t})$ , thanks to Corollary 1 and, therefore,  $\text{trc}(\sigma_{> t}, \tau_{\geq t}) \models \mathbf{dsc}(\varphi_1) \wedge \mathbf{dsc}(\varphi_2)$ . By the inductive hypothesis, it holds  $\sigma_{> t} \models \varphi_1 \wedge \varphi_2$ , for each such  $t$ . Lemma 5, then, gives us  $\sigma_{\geq t} \models \varphi_1 \wedge \varphi_2$ , for all  $t \in (t_i, t_{i+1})$ .

We now split the second part of the proof depending on whether  $\sigma$  is left-closed.

- **[ $\sigma$  is left-closed].** Take any  $t' \in J$ , where  $J = [t_0, t_i)$ , if  $\text{sing} \in \text{trc}(\sigma, \tau)_j$ , and  $J = [t_0, t_i]$ , otherwise. Clearly,  $\text{slice}_\sigma^\tau(t') \in \{0, \dots, j - 1\}$  and we have two cases, depending on whether  $t'$  is an element of  $\tau$  or lies in one of its open intervals. In the first case, let  $t_z \triangleq t'$  and  $z \triangleq \text{slice}_\sigma^\tau(t_z) < j$ . Since  $\text{trc}(\sigma, \tau)_{\geq z} \models \mathbf{dsc}(\varphi_1)$  and, by Corollary 1,  $\text{trc}(\sigma, \tau)_{\geq z} = \text{trc}(\sigma_{\geq t_z}, \tau_{\geq t_z})$ , we conclude  $\text{trc}(\sigma_{\geq t_z}, \tau_{\geq t_z}) \models \mathbf{dsc}(\varphi_1)$  and,

by the inductive hypothesis, also  $\sigma_{\geq t_z} \models \varphi_1$ . If, on the other hand,  $t' \in (t_l, t_{l+1})$ , for some  $l$ , let us set  $z \triangleq \text{slice}_\sigma^\tau(t') < j$ . Corollary 1 in this case gives us  $\text{trc}(\sigma, \tau)_{\geq z} = \text{trc}(\sigma_{> t'}, \tau_{\geq t'})$ . We know that  $\text{trc}(\sigma, \tau)_{\geq z} \models \mathbf{dsc}(\varphi_1)$ , hence,  $\text{trc}(\sigma_{> t'}, \tau_{\geq t'}) \models \mathbf{dsc}(\varphi_1)$ . By the inductive hypothesis,  $\sigma_{> t'} \models \varphi_1$  and, by Lemma 5, also  $\sigma_{\geq t'} \models \varphi_1$ .

- [ **$\sigma$  is left-open**]. For this case, the proof proceeds exactly as the previous one, where we consider the sets  $J = (t_0, t_i)$ , if  $\text{sing} \in \text{trc}(\sigma, \tau)_j$ , and  $J = (t_0, t_i]$ , otherwise, deriving  $\text{slice}_\sigma^\tau(t') \in \{1, \dots, j - 1\}$ .

Putting everything together, we have shown that there is a time  $t \in I$  such that  $\sigma_{\geq t} \models \varphi_2$  and  $\sigma_{\geq t'} \models \varphi_1$ , for all  $t' \in I$  with  $t' < t$ , which coincides with the semantic condition for  $\sigma \models \varphi_1 \cup \varphi_2$ . ■

We can now proceed to build an automaton that recognises all and only the discrete traces of the signals that satisfy a given RTL formula  $\varphi$ . First, define the following formulae, derived from  $\mathbf{dsc}(\varphi)$ :

$$\widehat{\varphi} \triangleq \mathbf{dsc}(\varphi) \wedge \mathbf{G}((\text{sing} \leftrightarrow \mathbf{X}\neg \text{sing}) \vee \text{last}) \quad (3)$$

$$\widehat{\varphi}^{\text{fin}} \triangleq \widehat{\varphi} \wedge \mathbf{F}(\text{last} \wedge \text{sing}). \quad (4)$$

Formula  $\widehat{\varphi}$  strengthens  $\mathbf{dsc}(\varphi)$  by requiring the alternation between singular and open observables, and formula  $\widehat{\varphi}^{\text{fin}}$  additionally stipulates that the trace is finite and corresponds to a right-closed signal.

We denote by  $\mathcal{A}_\varphi$  (resp.,  $\mathcal{A}_\varphi^{\text{fin}}$ ) the Büchi automaton (resp., finite automaton) for  $\widehat{\varphi}$  (resp.,  $\widehat{\varphi}^{\text{fin}}$ ) provided by Theorem 1, with components  $(2^{\widehat{AP}}, S, \delta, \lambda, S_0, S_F)$ . All components have their standard meaning. In particular,  $\lambda$  labels each state in  $S$  with a subset of  $\widehat{AP} = AP \cup \{\text{sing}\}$ . For convenience, we may write  $\llbracket s \rrbracket$  for  $\llbracket \lambda(s) \cap AP \rrbracket$  to denote the polyhedron interpreting the set of propositions labelling  $s$ .

As an immediate corollary of the above Theorem 2 and Proposition 5, we have the following theorem.

**Theorem 3.** *For all RTL formulae  $\varphi$ , the Büchi automaton  $\mathcal{A}_\varphi$  (resp., finite automaton  $\mathcal{A}_\varphi^{\text{fin}}$ ) satisfies the following property: for all infinite-time (resp., finite-time) signals  $\sigma$ , it holds that  $\sigma \models \varphi$  iff  $\text{trc}(\sigma, \tau)$  is recognised by  $\mathcal{A}_\varphi$  (resp.,  $\mathcal{A}_\varphi^{\text{fin}}$ ), for all time slicings  $\tau \in TS(\sigma)$ .*

Before concluding this section, it is worth observing that the automata constructed above have size only exponential in the length of the original RTL formula. Indeed, the Vardi–Wolper construction recalled in Theorem 1 yields an automaton whose size is exponential in the DAG-size of the input LTL formula, as its states correspond to sets of subformulae. Moreover, as noted earlier, the discretisation step introduces only a linear increase in DAG-size *w.r.t.* the original RTL formula, which is itself bounded by its length. The single-exponential claim therefore follows immediately.

## 5. Finite-Time Semantics

In the following for each scenario we define a finite abstraction of the model, called a *polyhedral abstraction*, *i.e.*, a finite graph whose nodes are labelled with a (possibly non-convex) polyhedron which is a subset of some combination of atomic propositions (*a.k.a.* an *observable*). Formally, a polyhedral abstraction is a tuple  $G = (V, E, pts, obs)$ , where  $(V, E)$  is a finite directed graph, and, for each node  $v \in V$ ,  $obs(v) \subseteq \widehat{AP}$  the set  $pts(v)$  is a non-empty polyhedron in  $\mathbb{R}^n$ ,  $obs(v) \subseteq \widehat{AP}$ , and it holds  $pts(v) \subseteq \llbracket obs(v) \cap AP \rrbracket$ . Since trajectories always end in a singular interval of a time slicing, we collect in the set  $\text{Pth}(G)$  (*resp.*, in  $\text{Pth}(G, k)$ ) the *sing-terminating paths* in the graph of  $G$  (*resp.*, the sing-terminating paths of length at most  $k$ ).

Polyhedral abstractions are meant to represent the relevant behaviours of polyhedral systems. To this aim, we introduce a relation between the two types of entities, called  $k$ -faithfulness, for a positive integer  $k$ . Roughly speaking, a polyhedral system is  $k$ -faithful to a polyhedral system if its paths of length  $k$  correspond to the length- $k$  discrete traces of the finite-time trajectories of the system.

The following definition uses  $TS(f, k)$  to denote the set of time slicings of  $f$  that induce a discrete trace of length at most  $k$ :

$$TS(f, k) \triangleq \{\tau \in TS(f) : |trc(f, \tau)| \leq k\},$$

or, equivalently,

$$TS(f, k) \triangleq \{t_0, \dots, t_\ell \in TS(f) \mid \ell \leq \lfloor k/2 \rfloor\}.$$

With an abuse of notation, we extend the function  $obs$  from nodes to paths in the polyhedral abstraction, returning the corresponding sequence of observables.

**Definition 2.** *Given an integer  $k \in \mathbb{N}$ , a polyhedral abstraction  $G = (V, E, pts, obs)$  is  $k$ -faithful iff the following conditions hold:*

1. *For all paths  $\pi \in \text{Pth}(G, k)$  and all points  $x \in pts(\text{first}(\pi))$  there exist a trajectory  $f \in \text{Traj}^{\text{fin}}(x)$  and a time slicing  $\tau \in TS(f, k)$  such that  $trc(f, \tau) = obs(\pi)$ .*
2. *For all points  $x \in \mathbb{R}^n$  and trajectories  $f \in \text{Traj}^{\text{fin}}(x)$ , if  $TS(f, k) \neq \emptyset$  then there is a time slicing  $\tau \in TS(f, k)$  and a path  $\pi \in \text{Pth}(G, k)$  s.t.  $x \in pts(\text{first}(\pi))$  and  $trc(f, \tau) = obs(\pi)$ .*

Roughly speaking, the first condition in the above definition ensures that each finite path of length at most  $k$  in the abstraction, starting from a node  $u$ , can be mimicked by a trajectory that starts from *any* point belonging to  $pts(u)$ . The second condition guarantees that the finite paths in the abstraction collectively cover all finite-time trajectories starting from all points in  $\mathbb{R}^n$ , up to the first  $k$  changes of observables.

Note that two nodes  $u$  and  $v$  may be labelled with the same observable and hence  $pts(u)$  and  $pts(v)$  may not be disjoint. Hence, a polyhedral abstraction does not induce an equivalence relation among points of  $\mathbb{R}^n$ . Rather, it puts together (*i.e.*, in the same set  $pts(u)$ ) *some* points that have *similar* trajectories.

*Sufficient horizon.* Thanks to the results in the previous section, if an ALTL formula  $\varphi$  is satisfied by a finite-time signal, then there is a finite trace induced by that signal that satisfies its discretisation  $\widehat{\varphi}^{\text{fin}}$ . The notion of sufficient horizon is meant to capture an upper bound on the length of the shortest such trace. Given an RTL formula  $\varphi$ , the sufficient horizon  $H_\varphi$  is equal to twice the number of states of  $\mathcal{A}_\varphi^{\text{fin}}$  times the maximum number of patches of any observable, namely

$$H_\varphi = 2 \cdot |S| \cdot K_{\max}, \quad \text{where } K_{\max} = \max_{\alpha \subseteq AP} |\text{Patch}(\alpha)|.$$

The following theorem states the main property of the sufficient horizon.

**Theorem 4.** *Given an RTL formula  $\varphi$  and a point  $x \in \mathbb{R}^n$ , if there is a trajectory  $f \in \text{Traj}^{\text{fin}}(x)$  whose signal satisfies  $\varphi$ , then there is a trajectory  $f' \in \text{Traj}^{\text{fin}}(x)$  s.t. its signal satisfies  $\varphi$  with a time slicing in  $TS(f', H_\varphi)$ .*

*Proof.* Assume that  $\sigma_f \models^{\text{fin}} \varphi$ , for some trajectory  $f \in \text{Traj}^{\text{fin}}(x)$ . Let  $f'$  be one of the trajectories with the shortest time slicing, among those trajectories whose signal satisfies  $\varphi$ . Formally, let  $F$  be the set of trajectories  $f \in \text{Traj}^{\text{fin}}(x)$  s.t.  $\sigma_f \models^{\text{fin}} \varphi$ , and let

$$f' \in \arg \min_{f \in F} \{|\tau| \mid \tau \in TS(f)\}.$$

Assume, by contradiction, that  $TS(f', H_\varphi)$  is empty, and let

$$\tau' = t_0, t_1, \dots, t_{\lfloor \ell/2 \rfloor} \in TS(f', \ell)$$

be one of the shortest time slicings of  $f'$ , with  $\ell > H_\varphi$ . Let  $\rho = s_0, s_1, \dots, s_{\ell-1}$  be an accepting run of  $\mathcal{A}_\varphi^{\text{fin}}$  on  $\text{trc}(\sigma_{f'}, \tau')$ . By Theorem 3, every run in  $\mathcal{A}_\varphi^{\text{fin}}$ ,  $\rho$  included, strictly alternates between states that are labelled with *sing* (*sing states*) and those that are not (*open states*). Hence,  $\rho$  contains  $\lfloor \ell/2 \rfloor$  open states. Clearly, any simple run of  $\mathcal{A}_\varphi^{\text{fin}}$  can at most contain  $|S|$  open states. Therefore, there must be an open state that occurs in  $\rho$  at least

$$\left\lceil \frac{\lfloor \ell/2 \rfloor}{|S|} \right\rceil > \max_{\alpha \subseteq AP} |\text{Patch}(\alpha)|$$

number of times, and let  $s$  be such an open state. By the above observations, there must be two indices  $0 \leq i < j < \ell$  such that  $(\rho)_i = (\rho)_j = s$  and  $f'(t)$  belongs to the same patch  $P$  of  $\llbracket s \rrbracket$ , for all  $t$  in a subinterval of the open interval  $(t_{\lfloor i/2 \rfloor}, t_{\lfloor i/2 \rfloor + 1})$  and also for all  $t$  in a subinterval of the open interval  $(t_{\lfloor j/2 \rfloor}, t_{\lfloor j/2 \rfloor + 1})$ . Let  $\widehat{t}_1$  be an arbitrary time instant in the first subinterval and  $\widehat{t}_2$  a time instant in the second one. By Lemma 1, we can replace the part of  $f'$  from  $\widehat{t}_1$  to  $\widehat{t}_2$  with a straight segment and obtain a trajectory  $f''$  that is still admissible, lies entirely within the convex polyhedron  $P$  during the

subinterval  $[\widehat{t}_1, \widehat{t}_2]$  and remains in the same observable  $\llbracket s \rrbracket$  containing  $P$  in the open interval  $(t_{\lfloor i/2 \rfloor}, t_{\lfloor j/2 \rfloor + 1})$ . Additionally, the signal  $\sigma_{f''}$  still satisfies  $\varphi$ , due to the shortened accepting run

$$\rho'' = s_0, s_1, \dots, s_i, s_{j+1}, \dots, s_{\ell-1}$$

of  $\mathcal{A}_\varphi^{\text{fin}}$  on  $\text{trc}(\sigma_{f''}, \tau'')$ . Moreover, the the shortened time slicing

$$\tau'' = t_0, t_1, \dots, t_{\lfloor i/2 \rfloor}, t_{\lfloor j/2 \rfloor + 1}, \dots, t_{\lfloor \ell/2 \rfloor}$$

belongs to  $TS(f'')$ . Therefore, the trajectory  $f'$  is not one of the trajectories with the shortest time slicing, which is a contradiction. Hence, the length of  $\tau'$  is at most  $H_\varphi$ , and this concludes the proof.  $\blacksquare$

Thanks to the above theorem, we can prove that the paths of a  $H_\varphi$ -faithful polyhedral abstraction are necessary and sufficient to characterise when a given point satisfies an RTL formula in the finite-time semantics.

**Theorem 5.** *Given an RTL formula  $\varphi$  and an  $H_\varphi$ -faithful polyhedral abstraction  $G$ , for all points  $x \in \mathbb{R}^n$ , the following are equivalent:*

- (i)  $x \models^{\text{fin}} \varphi$ ;
- (ii) there exist a node  $v$  and a path  $\pi \in \text{Pth}(G, H_\varphi)$  s.t.  $\text{first}(\pi) = v$ ,  $x \in \text{pts}(v)$ , and  $\text{obs}(\pi) \models \widehat{\varphi}^{\text{fin}}$ .

*Proof.* [(i) implies (ii)]. Let  $x \in \mathbb{R}^n$  be a point s.t.  $x \models^{\text{fin}} \varphi$ . By Theorem 4, there is a trajectory  $f \in \text{Traj}^{\text{fin}}(x)$  s.t.  $\sigma_f \models \varphi$  and  $TS(f, H_\varphi)$  is not empty. By Condition 2 of the definition of  $H_\varphi$ -faithful, there is a time slicing  $\tau \in TS(f, H_\varphi)$  and a path  $\pi$  in the polyhedral abstraction such that  $x$  belongs to  $\text{pts}(\text{first}(\pi))$  and  $\pi$  traverses the same observables as  $\text{trc}(\sigma_f, \tau)$ . Moreover, Theorem 3 guarantees that the discrete trace  $\text{trc}(\sigma_f, \tau)$  satisfies formula  $\widehat{\varphi}^{\text{fin}}$ . As a consequence,  $\text{obs}(\pi)$  satisfies  $\widehat{\varphi}^{\text{fin}}$ , which concludes the proof.

[(ii) implies (i)]. Let  $v \in V$  be a node and  $x \in \text{pts}(v)$  be a point, such that there exists a path  $\pi \in \text{Pth}(G, H_\varphi)$  that starts from  $v$  and satisfies  $\widehat{\varphi}^{\text{fin}}$  (i.e.,  $\text{obs}(\pi) \models \widehat{\varphi}^{\text{fin}}$ ). By Condition 1 of the definition of  $k$ -faithful, there is a trajectory  $f \in \text{Traj}^{\text{fin}}(x)$  and a time slicing  $\tau \in TS(f, H_\varphi)$  s.t.  $\text{trc}(f, \tau) = \text{obs}(\pi)$ . By Theorem 3,  $\sigma_f \models \varphi$  and hence  $x \models^{\text{fin}} \varphi$ .  $\blacksquare$

As a consequence of Theorem 5, having a  $H_\varphi$ -faithful polyhedral abstraction  $G$  is sufficient to solve the model-checking problem for  $\text{RTL}_f$ , by computing the *synchronous product* between the finite automaton  $\mathcal{A}_\varphi^{\text{fin}}$  and  $G$ , and then take the union of the denotations of all initial states of the product that can reach an accepting state. More precisely, let the components of the polyhedral abstraction be  $(V, E, \text{pts}, \text{obs})$ . Each state of the product automaton is a pair  $(s, v)$ , where  $s$  is a state of  $\mathcal{A}_\varphi^{\text{fin}}$  and  $v \in V$  is a node of the polyhedral abstraction. The transitions of the product automaton are obtained by synchronising the labels of the states of  $\mathcal{A}_\varphi^{\text{fin}}$  with the  $\text{obs}$  label from  $G$ . Initial and accepting states

are identified from  $\mathcal{A}_\varphi^{\text{fin}}$  based on the  $s$  component. The denotation of a product state  $(s, v)$  is taken from the polyhedral abstraction, *i.e.*, it is  $pts(v)$ .

In the following Sections 5.1 and 5.2, we describe two polyhedral abstractions, which respectively apply to the general case and to the special case of omnidirectional flow. Then, in Section 8 we discuss an on-the-fly algorithm that avoids the explicit computation of the product automaton.

### 5.1. The General Case

In order to define our first polyhedral abstraction, we introduce some auxiliary notation. We use the symbol  $\flat \in \{0, +\}$  to distinguish singular from open time intervals, and we denote by  $\neg\flat$  the symbol opposite to  $\flat$ . The states of the polyhedral abstraction correspond to finite sequences of patches of the observables, labelled with alternating 0's and +'s. More precisely, we assume a fixed representation of each observable  $\alpha \subseteq AP$  as a finite set of convex polyhedra  $\llbracket \alpha \rrbracket$ , and work with the set of patches

$$\mathcal{P} \triangleq \bigcup_{\alpha \subseteq AP} \llbracket \alpha \rrbracket.$$

For a patch  $P \in \mathcal{P}$ , we denote by  $\alpha_P$  the observable to which it belongs.<sup>4</sup> Let  $\mathcal{P}^*$  denote the set of finite sequences of patches and let  $\Delta \in \mathcal{P}^*$ , we define the function  $Trav^0(\Delta)$  (*resp.*,  $Trav^+(\Delta)$ ), to collect the left-closed (*resp.*, left-open) trajectories that *traverse*  $\Delta$ . Denoting by  $\varepsilon$  the empty sequence,  $Trav^\flat$  is recursively defined as follows:

$$\begin{aligned} Trav^\flat(\varepsilon) &= \mathbb{R}^n \\ Trav^0(P\Delta) &= reach^0(P, Trav^+(\Delta)) \\ Trav^+(P\Delta) &= P \cap reach^+(\llbracket \alpha_P \rrbracket, Trav^0(\Delta)). \end{aligned}$$

It is easy to see that  $Trav^\flat(\Delta)$  is a polyhedron, because the operators  $reach^0$  and  $reach^+$ , when applied to polyhedra, return other polyhedra.

As an example, consider the set

$$Trav^0(PQR) = reach^0(P, Q \cap reach^+(\llbracket \alpha_Q \rrbracket, reach^0(R, \mathbb{R}^n))).$$

It collects points of  $P$  from which there is a trajectory that immediately moves to  $Q$ , and then reaches  $R$  after spending some time in the observable of  $Q$  (possibly passing through several other patches of that observable). The two  $Trav$  operators are instrumental in the following definition of a polyhedral abstraction.

---

<sup>4</sup>Recall that the denotations of different observables are disjoint, so every patch in  $\mathcal{P}$  belongs to exactly one observable.

*The polyhedral abstraction.* Our first polyhedral abstraction  $Fin(\mathcal{P}, H) = (V, E, pts, obs)$  is built from a polyhedral system  $\mathcal{P}$  and a sufficient horizon  $H > 0$ , and comprises the following components:

- **Nodes:**  $V$  contains a node for each pair  $(b, \Delta)$ , where  $b \in \{0, +\}$  and  $\Delta$  is a finite non-empty sequence of patches of observables, of length at most  $H$ . Additionally, to be in  $V$  a node must have a non-empty denotation according to  $pts$  (see below).
- **Edges:** For all nodes  $(b, P\Delta)$ , there is an edge to  $(\neg b, \Delta)$ .
- **Observables:**  $obs((0, P\Delta)) = \alpha_P \cup \{sing\}$  and  $obs((+, P\Delta)) = \alpha_P$ .
- **Points:** The denotation of  $(0, \Delta)$  (resp.,  $(+, \Delta)$ ) is the set of points from which there is a left-closed (resp., left-open) trajectory that traverses the sequence  $\Delta$ . Formally,  $pts((b, \Delta)) = Trav^b(\Delta)$ .

Observe that, in general, the denotations of the nodes are not disjoint. Note that the resulting graph  $(V, E)$  is a forest with edges from each node to its parent. The roots of this forest are the nodes containing sequences of length 1.

**Example 1.** *Let us analyse the polyhedral abstraction corresponding to the 2-tank example in the Introduction, assuming a sufficient horizon  $H$  greater than or equal to 5. Consider again the formula*

$$\varphi_1^{\text{gap}} = (t = 0) \wedge G(t \leq 10) \wedge F(a \geq b + 1 \wedge F(b \geq a + 1)).$$

The formula contains the following four atomic propositions:

Proposition	Denotation $[\cdot]$
$p_a$	$a \geq b + 1$
$p_b$	$b \geq a + 1$
$T_0$	$t = 0$
$T_{\leq 10}$	$t \leq 10$

By taking subsets of atomic propositions, we obtain  $2^4 = 16$  observables. However, only 9 of them correspond to non-empty denotations:

$$\{\}, \{T_{\leq 10}\}, \{T_0, T_{\leq 10}\}, \{p_a\}, \{p_a, T_{\leq 10}\}, \{p_a, T_0, T_{\leq 10}\}, \{p_b\}, \{p_b, T_{\leq 10}\}, \{p_b, T_0, T_{\leq 10}\}$$

For instance, the denotation of the empty observable is  $[\{\}] = \{t > 10, a < b + 1, b < a + 1\}$ , which is a convex polyhedron in  $\mathbb{R}^3$ . In fact, it is easy to check that in this example the denotations of all observables are convex polyhedra, so each of them contains a single patch.

Now, we describe a fragment of the polyhedral abstraction  $Fin(\mathcal{P}, H)$ . Intuitively, nodes of the polyhedral abstraction correspond to sets of (similar) trajectories of the original polyhedral system. Consider the two left-closed finite-time trajectories  $f_1, f_2$  whose projection on the  $(a, b)$ -plane is depicted in Figure 5. The first trajectory starts from the initial point  $(t, a, b) = (0, 5, 5)$ , proceeds along the direction  $(\dot{t}, \dot{a}, \dot{b}) = (1, 1, 1)$  for 3 time units and ends at  $(3, 8, 8)$ , thus traversing the following observables:  $\{T_0, T_{\leq 10}\}, \{T_{\leq 10}\}$ . Next, the trajectory  $f_2$

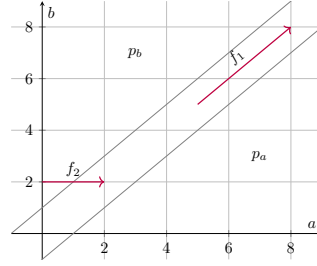


Figure 5: Two trajectories from Example 1, projected on variables  $a, b$ .

starts from  $(0, 0, 2)$  and ends at  $(2, 2, 2)$ , following the direction  $(1, 1, 0)$  for 2 time units, traversing the observables  $\{p_b, T_0, T_{\leq 10}\}, \{p_b, T_{\leq 10}\}, \{T_{\leq 10}\}$ . These two trajectories are represented by the fragment of polyhedral abstraction in Figure 6. The trajectory  $f_1$  corresponds to a path of length three in the polyhedral abstraction, because it starts from the observable  $\{T_0, T_{\leq 10}\}$ , immediately moves to  $\{T_{\leq 10}\}$ , where it spends three time units, and then ends in that same observable. The trajectory  $f_2$ , instead, requires a path of length five, because it crosses three observables and spends time in two of them.

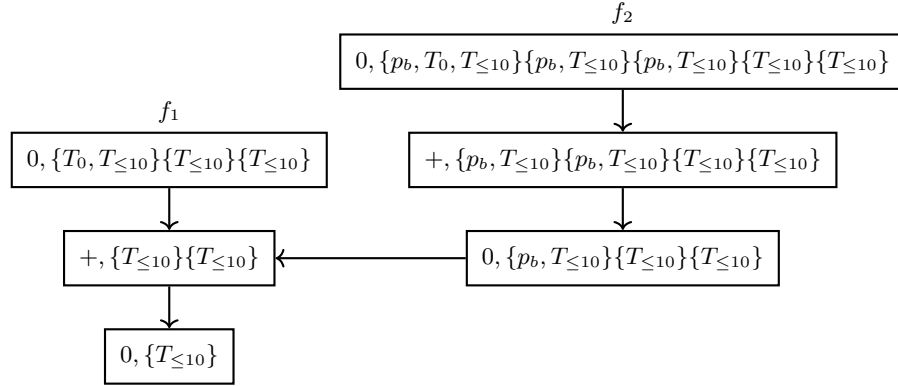


Figure 6: A fragment of the polyhedral abstraction  $Fin(\mathcal{P}, H)$  (see Example 1).

**Theorem 6.** *The above polyhedral abstraction  $G = Fin(\mathcal{P}, H_\varphi)$  is  $H_\varphi$ -faithful.*

*Proof.* [Condition (1)] Let  $\pi \in \text{Pth}(G, H_\varphi)$  be a path of length  $\ell \leq H_\varphi$ , ending in a *sing* observable. Let  $u = (b, \Delta)$  be the first node of  $\pi$ , and  $x \in \mathbb{R}^n$  be a point s.t.

$$x \in \text{pts}(u) = \text{Trav}^b(\Delta).$$

Since  $\pi$  starts from  $u$  and is long  $\ell$ , by definition of  $G$  the sequence  $\Delta$  contains  $k \geq \ell$  convex polyhedra. Then, by definition of  $\text{Trav}^b$  there is a trajectory  $f \in \text{Traj}^{\text{fin}}(x)$  that traverses  $\Delta$ , and consequently a time slicing

$$\tau = t_0, t_1, \dots, t_{\lfloor k/2 \rfloor} \in \text{TS}(f, k).$$

In particular, the discrete trace  $trc(f, \tau)$ , if restricted to its first  $\ell$  symbols, coincides with  $obs(\pi)$ . By truncating  $f$  and  $\tau$  at  $t_{\lfloor \ell/2 \rfloor}$ , we obtain a trajectory  $f'$  and a time slicing  $\tau'$  s.t.  $trc(f', \tau') = obs(\pi)$ , as required.

**[Condition (2)]** Let  $x \in \mathbb{R}^n$  be a point,  $f \in Traj^{\text{fin}}(x)$  be a trajectory starting from  $x$  and let

$$\tau = t_0, t_1, \dots, t_\ell \in TS(f, H_\varphi),$$

for some  $\ell \leq \lfloor H_\varphi/2 \rfloor$ . Assume for simplicity that  $f$  is left-closed, and let  $\beta_0\beta_1 \dots \beta_{2\ell-1} = trc(f, \tau)$ . For  $i = 0, 1, \dots, \ell$ , let  $P_{2 \cdot i}$  be the patch of  $\beta_{2 \cdot i} \cap AP$  where  $f(t_i)$  lies. For  $i = 0, 1, \dots, \ell - 1$ , let  $P_{2 \cdot i+1}$  be the *first* patch of  $\beta_{2 \cdot i+1} \cap AP$  that  $f$  encounters during the interval  $(t_i, t_{i+1})$ .<sup>5</sup> Let  $\Delta = P_0P_1 \dots P_{2\ell-1}$ , it is easy to see that  $x \in Trav^0(\Delta)$ . Hence, the pair  $(0, \Delta)$  is a node in the polyhedral abstraction, and from that node starts a path

$$\pi = (0, \Delta) (+, P_1P_2 \dots P_{2\ell-1}) (0, P_2P_3 \dots P_{2\ell-1}) \dots (0, P_{2\ell-1}),$$

whose sequence of observables is equal to  $trc(f, \tau)$ , which concludes the proof. ■

The first line in Table 2 reports the main characteristics of the present scenario. The number of states in the polyhedral abstraction is the number of all sequences of patches, from any observable, of length at most equal to the sufficient horizon  $H_0 = 2 \cdot 2^{|\varphi|} \cdot K_{\max}$ . The overall complexity follows from the size of the product automaton, bounded by  $2^{|\varphi|} \cdot K^{H_0+1}$ , which is  $K^{O(2^{|\varphi|} \cdot K_{\max})}$ . Finding the initial states that can reach an accepting state can be performed in linear time. The union of their denotations answers the existential denotation problem, as discussed on Page 27.

### 5.2. A Special Case: Omnidirectional Flow

When the origin  $\mathbf{0}$  belongs to the interior of  $Flow$ , the problem becomes simpler and a much more succinct abstraction based in geometric adjacency suffices. We shall call *tile* a maximal connected set of patches of the same observable contained in the invariant. Every tile is, therefore, a polyhedron. Despite the fact that tiles are not necessarily convex, every point  $x$  of a tile is reachable from any other point  $y$  in same tile by means of an admissible trajectory that never exits the tile. Formally:

**Lemma 8.** *Assume that  $\mathbf{0} \in \text{int}(Flow)$ . Let  $P$  be a tile and  $x$  and  $y$  be any two points in  $P$ . Then there exists an admissible trajectory  $f$  and a time delay  $\delta \geq 0$  such that  $f(0) = x$ ,  $f(\delta) = y$ , and  $f(t) \in P$ , for all  $t \in (0, \delta)$ .*

*The polyhedral abstraction.* Our second polyhedral abstraction  $\text{Omni}(\mathcal{P}) = (\mathbb{V}, \mathbb{E}, \text{pts}, \text{obs})$  comprises the following components:

<sup>5</sup>This is well defined, because we are assuming that trajectories are well-behaved.

- **Nodes:** We have a node  $v = (P)$  for each tile  $P$  and a node  $v = (P, Q, R)$ , for each triple of tiles  $P$ ,  $Q$ , and  $R$ , provided its denotation  $pts(v)$ , defined below, is non-empty.
- **Edges:** If the nodes  $(P)$  and  $(P, Q, R)$  both belong to  $V$ , then the pair  $((P), (P, Q, R))$  belongs to  $E$ . Moreover, if  $(R)$  and  $(P, Q, R)$  both belong to  $V$ , then the pair  $((P, Q, R), (R))$  belongs to  $E$ .
- **Observables:** For each node  $v \in V$ , if  $v$  is of the form  $(P)$ , then  $obs(v)$  is the set of atomic propositions true at the points in  $P$ , while if  $v$  is of the form  $(P, Q, R)$ , then  $obs(v)$  is the set of atomic propositions true at the points in  $Q$  together with the proposition *sing*.
- **Points:** The denotation  $pts((P))$  of the node  $(P)$  is the set of points contained in the tile  $P$  itself, while  $pts((P, Q, R))$  is the set of points of the tile  $Q$  that can reach immediately both  $P$  and  $R$ ; that is, those points  $x \in Q$  with the following property: there exist two points  $y \in P$  and  $z \in R$ , two admissible trajectories  $f_1$  and  $f_2$ , and two time instants  $t_1$  and  $t_2$  such that  $f_1(0) = f_2(0) = x$ ,  $f_1(t_1) = y$ ,  $f_2(t_2) = z$ ,  $f_1(t) \in P$ , for all  $t \in (0, t_1)$ , and  $f_2(t) \in R$ , for all  $t \in (0, t_2)$ . Since  $\mathbf{0} \in int(Flow)$ , any straight segment is an admissible trajectory and the above property becomes equivalent to requiring that point  $x$  be adjacent to both  $P$  and  $R$ . Therefore, the set  $pts((P, Q, R))$  corresponds precisely to  $Q \cap cl(P) \cap cl(R)$ . As already said, the set of nodes  $V$  contains all and only the nodes with non-empty denotation.

The intuitive idea here is that a path in  $Omni(\mathcal{P})$ , which strictly alternates between nodes of the form  $(P)$  and nodes of the form  $(P, Q, R)$ , corresponds to an abstract representation of pairs  $(f, \tau)$  of a trajectory and a time slicing, where a node  $(P)$  captures a region of some observable in which  $f$  resides in an open interval of  $\tau$ , while a node  $(P, Q, R)$  represents instantaneous passages corresponding to a singular interval in  $\tau$ . Note that nodes of the form  $(P, P, P)$  are permitted and necessary to allow for stuttering time slicings, where a trajectory is contained in the denotation of the same observable for multiple consecutive time intervals of a time slicing.

**Example 2.** *The two-tank example from the introduction falls in the omnidirectional flow case, so we can use it again to exemplify the second polyhedral abstraction. To this aim, consider again the two trajectories  $f_1$  and  $f_2$  from Example 1. Figure 7 shows the fragment of polyhedral abstraction corresponding to  $f_1$  and  $f_2$ . The dashed arrow and node belong to the polyhedral abstraction, but are not involved in the trajectories  $f_1$  and  $f_2$ .*

*The six nodes representing the two trajectories can similarly be identified in Figure 6. However, the sets of points  $pts(v)$  associated to each node  $v$  in the two graphs differ. For instance, consider the two nodes labelled with  $f_2$  in the two figures:*

$$\text{Fig. 6} \quad v = (0, \{p_b, T_0, T_{\leq 10}\} \{p_b, T_{\leq 10}\} \{p_b, T_{\leq 10}\} \{T_{\leq 10}\} \{T_{\leq 10}\})$$

$$\text{Fig. 7} \quad v' = (\{p_b, T_0, T_{\leq 10}\}, \{p_b, T_0, T_{\leq 10}\}, \{p_b, T_{\leq 10}\})$$

*Both  $pts(v)$  and  $pts(v')$  contain the starting point of the trajectory  $f_2$ , namely*

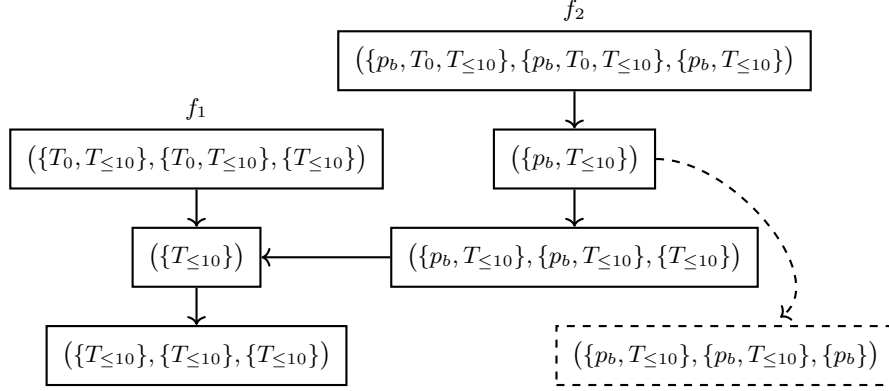


Figure 7: A fragment of the polyhedral abstraction  $\text{Omni}(\mathcal{P})$  (see Example 2).

$(t, a, b) = (0, 0, 2)$ ; however,  $\text{pts}(v)$  is strictly contained in  $\text{pts}(v')$ . Indeed,  $\text{pts}(v)$  contains only points satisfying  $p_b$  that can exit from  $p_b$  within 10 time units, whereas  $\text{pts}(v')$  does not impose this constraint. For instance, the point  $(0, 0, 20)$  belongs to  $\text{pts}(v') \setminus \text{pts}(v)$ .

**Theorem 7.** *The polyhedral abstraction  $G = \text{Omni}(\mathcal{P})$  is  $k$ -faithful for all integers  $k \geq 1$ .*

*Proof.* In order to prove correctness of the polyhedral abstraction we need to show that (i) every path in the graph underlying  $G$  is witnessed by some trajectory and time-slicing (Item 1 of Definition 2), and that (ii) for every point  $x$  and trajectory  $f$  from  $x$ , there exists a path in  $G$  starting from a vertex containing  $x$  that exhibits the same observables as the trace induced by  $f$  (Item 2 of Definition 2).

- (i) Let us start by showing that the first item of Definition 2 is satisfied for every  $k \geq 1$ . We proceed by induction on  $k$  and prove that, for every  $\pi \in \text{Pth}(G, k)$  and  $x$  a point in  $\text{pts}(\text{first}(\pi))$ , there exists a trajectory  $f \in \text{Traj}^{\text{fin}}(x)$  and a time slicing  $\tau \in \text{TS}(f, k)$  such that  $\text{trc}(f, \tau) = \text{obs}(\pi)$  and, in addition, that  $f$  is left-open if  $\text{first}(\pi)$  is of the form  $(P)$ , for some tile  $P$ , while it is left-closed if  $\text{first}(\pi) = (P, Q, R)$ , for some tiles  $P, Q$  and  $R$ .

The base case is for  $k = 1$  (the path only consists of a single node) and it can only be of the form  $\pi = (P, Q, R)$ , for some tiles  $P, Q$  and  $R$ , where  $\text{pts}((P, Q, R)) \neq \emptyset$ . Then, for every point  $x \in \text{pts}((P, Q, R))$ , we set  $f$  to be the left-closed trajectory defined on the singular interval  $(0, 0)$  by  $f(0) = x$  and  $\tau = 0 \in \text{TS}(f, 1)$ . Since  $\text{pts}((P, Q, R))$  is all contained in  $Q$  and the points of  $Q$  are all contained in the denotation of the the same observable, it is immediate to conclude that  $\text{trc}(f, \tau) = \text{obs}(\pi)$ .

For the inductive case with  $k > 1$ , let us consider first the case  $\pi = (P) \cdot \pi'$ , where  $\pi' \in \text{Pth}(G, k - 1)$  and starts with a node of the form  $(P, Q, R)$ , for

some tiles  $Q$  and  $R$ . If the length of  $\pi$  is strictly less than  $k$ , the inductive hypothesis already gives us the result. Assume, then, that its length be exactly  $k$ . In this case, the  $k$  is necessarily even, since each path must strictly alternate between the two types of nodes and must end in a node of the form  $(P', Q', R')$ . Let  $x$  be any point in  $pts((P))$ . Since  $\mathbf{0} \in \text{int}(Flow)$ , by Lemma 8, any point in  $pts((P, Q, R))$  can reach any point in  $(P)$  and *vice versa*. Hence, from  $x$  there is a left-open trajectory  $\bar{f}$  and an interval  $(0, \bar{t}]$  such that  $\bar{f}(0) = x$ ,  $y \triangleq \bar{f}(\bar{t}) \in pts((P, Q, R))$  and  $\bar{f}(t) \in pts((P))$ , for all  $t \in (0, \bar{t})$ . By induction hypothesis, there exist a left-closed trajectory  $f' \in Traj^{\text{fin}}(y)$  and a time slicing  $\tau' = 0, t'_1, \dots, t'_l \in TS(f, k-1)$  such that  $\text{trc}(f', \tau') = \text{obs}(\pi')$ . Observe that, this last equality ensures that the trace induced by  $\tau'$  must be of the same length  $k-1$  as the path  $\pi'$ , and, therefore, it must be the case that  $l = \lfloor \frac{k-1}{2} \rfloor$ . By adding 0 to a version of  $\tau'$  shifted forward by  $\bar{t}$  and concatenating  $\bar{f}$  with  $f'$ , we build the desired time slicing  $\tau$  and left-open trajectory  $f$ . Formally,  $\tau \triangleq 0, t_1, \dots, t_{\lfloor \frac{k-1}{2} \rfloor + 1}$ , where  $t_i = \bar{t} + t'_{i-1}$ , for all  $i \in [1, \lfloor \frac{k-1}{2} \rfloor + 1]$ . Since  $k$  is even,  $\lfloor \frac{k-1}{2} \rfloor = \lfloor \frac{k}{2} \rfloor - 1$ , hence,  $\lfloor \frac{k-1}{2} \rfloor + 1 = \lfloor \frac{k}{2} \rfloor$ , and, therefore,  $\tau \in TS(f, k)$ . Then, we set

$$f(t) = \begin{cases} \bar{f}(t) & \text{if } t \in (0, \bar{t}) \\ f'(t - \bar{t}) & \text{if } t \in [\bar{t}, t_{\lfloor \frac{k-1}{2} \rfloor + 1}] \end{cases}.$$

Since  $\text{trc}(\bar{f}, (0, t_1)) = \text{obs}((P))$ , it is immediate to conclude that  $\text{trc}(f, \tau) = \text{obs}(\pi)$ .

Assume, now, that  $\pi = (P, Q, R) \cdot \pi'$ , for some tiles  $P, Q$  and  $R$ , where  $\pi' \in \text{Pth}(G, k-1)$ , starts with the node  $(P)$  and is of length exactly  $k$  (otherwise the induction hypothesis is already enough). Let  $x \in pts((P, Q, R)) \subseteq Q$ . By definition of  $pts((P, Q, R))$ , there exists a left-open trajectory  $\bar{f} \in Traj^{\text{fin}}(x)$  and a time instant  $\bar{t} > 0$  with  $\bar{f}(t) \in P$ , for all  $t \in (0, \bar{t}]$ . By induction hypothesis, there exist a left-open trajectory  $f' \in Traj^{\text{fin}}(\bar{f}(\bar{t}))$  and a time slicing  $\tau' = 0, t'_1, \dots, t'_l$  satisfying  $\text{trc}(f', \tau') = \text{obs}(\pi')$  and, therefore,  $l = \lfloor \frac{k-1}{2} \rfloor$ . Let us define a left-closed trajectory  $f \in Traj^{\text{fin}}(x)$  as follows:

$$f(t) = \begin{cases} x & \text{if } t = 0 \\ \bar{f}(t) & \text{if } t \in (0, \bar{t}] \\ f'(t - \bar{t}) & \text{if } t \in (\bar{t}, t_{\lfloor \frac{k-1}{2} \rfloor}] \end{cases}.$$

and let  $\tau \triangleq 0, t_1, \dots, t_{\lfloor \frac{k-1}{2} \rfloor}$  be such that  $t_i = \bar{t} + t'_i$ , for all  $i \in [1, \lfloor \frac{k-1}{2} \rfloor]$ .

Since  $k$  in this case is necessarily odd,  $\lfloor \frac{k-1}{2} \rfloor = \lfloor \frac{k}{2} \rfloor$ . In addition,  $f(0)$  lies in  $Q$ ,  $f(t)$  lies in the same tile  $P$ , hence in the same observable, for all  $t \in (0, t_1)$ . As a consequence,  $\tau \in TS(f, k)$  and  $\text{trc}(f, \tau) = \text{obs}(\pi)$ .

- (ii) Let us now consider Item 2 of Definition 2. Again, we proceed by induction on  $k \geq 1$  and consider an arbitrary point  $x \in P$ , for some tile  $P$ , and a

trajectory  $f$  from  $x$  and build the path in  $G$  with the exhibiting the same sequence of observables.

The base case for  $k = 1$ ,  $f(0) = x$ , and  $\tau = 0$  (observe that for  $k = 1$  those are the only possible trajectory and time slicing). The corresponding path  $\pi \in \text{Pth}(G, 1)$  contains the single node  $(Q, P, R)$ , for some  $Q$  and  $R$ , and  $\text{trc}(f, \tau) = \text{obs}((Q, P, R))$ .

For the inductive step, let us first consider the case of a left-closed finite-time trajectory  $f$  with a time slicing  $\tau \in TS(f, k)$ . Assume that  $\tau = 0, t_1, \dots, t_{\lfloor \frac{k}{2} \rfloor}$  (if  $\tau$  is shorter the induction hypothesis immediately gives the result). Let  $P$  be tile to which the patch containing  $f(0) = x$  belongs. By definition of  $\text{trc}(f, \tau)$ , all points  $f(t)$ , for  $t$  in  $(0, t_1)$ , must be contained in the denotation of the same observable  $\alpha_0$ . Let  $Q$  be the tile contained in the denotation of  $\alpha_1$  and that contains all such points (one such tile must exist since  $f$  never leaves the denotation of  $\alpha_1$  in  $(0, t_1)$ ). Then, the denotations of the nodes  $(Q)$  and  $(Q, P, Q)$  are not empty and, therefore,  $(Q, P, Q), (Q) \in V$ . Moreover, there is an edge in  $E$  from  $(Q, P, Q)$  to  $(Q)$ , by definition of the polyhedral abstraction. Let now  $f'$  be the left-open trajectory that coincides with  $f$  on all the time instants  $t > 0$ , and let  $\tau' \triangleq \tau$ . Since in this case  $k$  is odd,  $\lfloor k/2 \rfloor = \lfloor (k-1)/2 \rfloor$  and, therefore,  $\tau'$  also belongs to  $TS(f', k-1)$ . By induction hypothesis, there exists a path  $\pi' \in \text{Pth}(G, k-1)$ , with  $\text{trc}(f', \tau') = \text{obs}(\pi')$  and  $(Q)$  as first node in  $\pi'$ . Since the length of  $\text{trc}(f', \tau')$  is the same as that of  $\text{obs}(\pi')$ , which is  $k-1$ , the path  $\pi \triangleq (P, Q, P) \cdot \pi'$  is a path of length  $k$  in  $\text{Pth}(G, k)$  and  $\text{trc}(f, \tau) = \text{obs}(\pi)$  as required.

Consider now the case of a left-open trajectory  $f$  from  $x$  and a time slicing  $\tau = 0, t_1, \dots, t_{\lfloor \frac{k}{2} \rfloor} \in TS(f, k)$ . Then,  $f(t) \in P$  (and  $\lim_{t \rightarrow 0} f(t) = x$ ), for all  $t \in (0, t_1)$ , where  $P$  is the tile to which the observable  $\alpha_0$  satisfied by  $f$  in the interval  $(0, t_1)$  belongs. Consider the following time slicing  $\tau' \triangleq 0, t'_1, \dots, t'_{\lfloor \frac{k}{2} \rfloor - 1}$ , where  $t'_i = (t_{i+1} - t_1)$ , for all  $i \in [1, \lfloor k/2 \rfloor - 1]$ , and the left-closed trajectory  $f'$  defined by  $f'(t) = f(t + t_1)$ , for all  $t \in [0, (t_{\lfloor \frac{k}{2} \rfloor} - t_1)]$ . Since  $k$  is even,  $\lfloor \frac{k}{2} \rfloor - 1 = \lfloor \frac{k-1}{2} \rfloor$  and, therefore,  $\tau' \in TS(f', k-1)$ . By the induction hypothesis, there exists a path  $\pi' \in \text{Pth}(G, k-1)$ , with  $\text{trc}(f', \tau') = \text{obs}(\pi')$  that starts with node  $(P, Q, P)$ , for some tile  $Q$  containing the point  $f'(0) = f(t_1)$ . Since all the points  $f(t)$ , with  $t \in (0, t_1)$ , satisfy  $\text{obs}((P))$  and, by the definition of the polyhedral abstraction, there exists an edge from  $(P)$  to  $(P, Q, P)$ , the path  $\pi \triangleq (P) \cdot \pi'$  belongs to  $\text{Pth}(G, k)$  and is such that  $\text{trc}(f, \tau) = \text{obs}(\pi)$ . ■

The second line in Table 2 reports the main characteristics of the present scenario. The number of states in the polyhedral abstraction is the number of tiles of all observables, and of triples of those same tiles. Since tiles are fewer than patches, we obtain the bound  $|V| \leq K^3 + K$ . The overall complexity follows from the size of the product automaton, bounded by  $2^{|\varphi|} \cdot (K^3 + K)$ , which is

$O(K^3 \cdot 2^{|\varphi|})$ . Finding the initial states that can reach an accepting state can be performed in linear time.

## 6. Infinite-time Semantics

### 6.1. First Scenario: Omnidirectional Flow

In order to tackle the infinite-time scenarios for omnidirectional flow, we need to lift to the infinite-time case the notion of  $k$ -faithful abstraction given in Definition 2 for the finite-time case. This corresponds to the notion of *faithful abstraction* as defined below.

**Definition 3.** A polyhedral abstraction  $G = (V, E, pts, obs)$  is faithful iff the following conditions hold:

1. for all infinite paths  $\pi \in \text{Pth}(G)$  and points  $x \in pts(\text{first}(\pi))$ , there exist a trajectory  $f \in \text{Traj}^{\text{inf}}(x)$  and a time slicing  $\tau \in \text{TS}(f)$  such that  $\text{trc}(f, \tau) = \text{obs}(\pi)$ ;
2. for all points  $x \in \mathbb{R}^n$ , trajectories  $f \in \text{Traj}^{\text{inf}}(x)$ , and time slicings  $\tau \in \text{TS}(f)$ , there exists an infinite path  $\pi \in \text{Pth}(G)$  such that  $\text{trc}(f, \tau) = \text{obs}(\pi)$ .

The following theorem states that, as far as  $\text{RTL}_\omega$  satisfaction is concerned, faithful polyhedral abstractions are sound and complete abstractions of polyhedral systems under the infinite-time semantics.

**Theorem 8.** For all faithful polyhedral abstractions  $G$ , and RTL formulae  $\varphi$ :

- (i) for all nodes  $v \in V$  and points  $x \in pts(v)$ , if there exists an infinite path in  $\text{Pth}(G)$  that starts from  $v$  and satisfies  $\hat{\varphi}$  then  $x \models^{\text{inf}} \varphi$ ;
- (ii) for all points  $x \in \mathbb{R}^n$  such that  $x \models^{\text{inf}} \varphi$ , there exists an infinite path in  $\text{Pth}(G)$  that satisfies  $\hat{\varphi}$  and starts from a node  $v$  with  $x \in pts(v)$ .

*Proof.* Let us first consider Item 1 of the statement above. Take an infinite path  $\pi$  of  $G$  and an arbitrary point  $x$  in  $pts(\text{first}(\pi))$ . By assumption,  $\pi$  satisfies  $\hat{\varphi}$ , which formally means that  $\text{obs}(\pi) \models \hat{\varphi}$ . By Item 1 of Definition 3 of faithful abstraction, there exist a trajectory  $f \in \text{Traj}^{\text{inf}}(x)$  and a time slicing  $\tau \in \text{TS}(f)$ , with  $\text{trc}(\sigma_f, \tau) = \text{trc}(f, \tau) = \text{obs}(\pi)$ , where we recall that  $\sigma_f$  denotes the signal of  $f$ . Therefore, we also have that  $\text{trc}(\sigma_f, \tau) \models \hat{\varphi}$ . The conclusion, now, immediately follows from Theorem 3, which ensures that  $\sigma_f \models \varphi$ . Signal  $\sigma_f$  itself is a witness for  $x \models^{\text{inf}} \varphi$ .

As to Item 2, let us take any point  $x \in \mathbb{R}^n$  such that  $x \models^{\text{inf}} \varphi$ . This means that there exists a trajectory  $f$  in  $\text{Traj}^{\text{inf}}(x)$  whose signal  $\sigma_f$  satisfies  $\sigma_f \models \varphi$ . By Theorem 3,  $\sigma_f \models \varphi$  implies that  $\text{trc}(\sigma_f, \tau) \models \hat{\varphi}$ , for any time slicing  $\tau \in \text{TS}(f)$ . By Item 2 of Definition 3, there exists an infinite path  $\pi \in \text{Pth}(G)$  with  $\text{trc}(\sigma_f, \tau) = \text{trc}(f, \tau) = \text{obs}(\pi)$ . Therefore  $\text{obs}(\pi) \models \hat{\varphi}$  as asserted by the statement. ■

The polyhedral abstraction of Section 5.2 turns out to be adequate also for infinite-time trajectories, as the following theorem states.

**Theorem 9.** *The polyhedral abstraction of Section 5.2 is faithful.*

*Proof.* Let us consider Item 1 of Definition 3 and let

$$\pi = (Q_0) (Q_0, P_0, Q_1) (Q_1) (Q_1, P_1, Q_2) \cdots (Q_n) (Q_n, P_n, Q_{n+1}) \cdots$$

be an infinite path in  $\text{Pth}(G)$  and  $x$  an arbitrary point in  $\text{pts}((Q_0)) = Q_0$ . For this case, we shall build a left-open infinite-time trajectory  $f$  and an infinite time slicing  $\tau$  for  $f$  whose trace is identical to the sequence of observations of  $\pi$ , i.e.,  $\text{trc}(f, \tau) = \text{obs}(\pi)$ . Let  $\{x_i\}_{i \geq 0}$  be any infinite sequence of points such that  $x_i \in \text{pts}((Q_i, P_i, Q_{i+1}))$ . Clearly, each  $x_i$  lies on the border between  $Q_i$  and  $P_i$  and also on the border between  $P_i$  and  $Q_{i+1}$ . By applying Lemma 8, we obtain a finite-time trajectory  $f_0$  leading from  $x$  to  $x_0$  and a delay  $\delta_0$  such that  $f_0(\delta_0) = x_0$  as well as, for each  $i \geq 1$ , a trajectory  $f_i$  from  $x_{i-1}$  to  $x_i$  and a corresponding delay  $\delta_i$  such that  $f_i(\delta_i) = x_i$ . Observe that, for each  $i \geq 0$ , we have  $f_i(t) \in Q_i$ , for all  $t \in (0, \delta_i)$ . We define  $\tau = \{t_i\}_{i \geq 0}$  so that  $t_i = \sum_{j=0}^{i-1} \delta_j$ , for all  $i \geq 0$ . Moreover, let  $f$  be the trajectory defined as follows: for all  $i \geq 0$  and all  $t \in (\delta_i, \delta_{i+1}]$ , we set  $f(t) = f_i(\delta_i + t)$ . By construction,  $f \in \text{Traj}^{\text{inf}}(x)$  and  $\tau \in \text{TS}(f)$ . Indeed, in any open interval of  $\tau$  the trajectory  $f$  is contained in the same tile, hence the observable in all those points is the same. Due to the way in which we built  $f$  and  $\tau$  based on  $\pi$ , it follows immediately that  $\text{trc}(f, \tau) = \text{obs}(\pi)$ . The case in which  $\pi$  starts with a node of the form  $(Q_0, P_0, Q_1)$  is very similar, leading to a left-closed infinite-time trajectory  $f$  and a time slice  $\tau$  with the same properties as those discussed above. The proof for this case is, thus, omitted.

Let us consider Item 2 of Definition 3 and let  $x \in \mathbb{R}^n$  be an arbitrary point,  $f \in \text{Traj}^{\text{inf}}(x)$  an infinite-time left-closed trajectory from  $x$ . Since  $f$  is well-behaved, the set  $\text{TS}(f)$  of time slicings for  $f$  is non-empty. Then, let  $\tau = \{t_i\}_{i \geq 0} \in \text{TS}(f)$  be an arbitrary infinite time slicing for  $f$ , we have:

$$\text{trc}(f, \tau) = \beta_0 \alpha_0 \beta_1 \alpha_1 \cdots \beta_n \alpha_n \cdots$$

for some infinite sequence  $\beta_0 \alpha_0 \beta_1 \alpha_1 \cdots \beta_n \alpha_n \cdots$  of observables. Since  $f$  is continuous, there must be a corresponding sequence of adjacent (and not necessarily all different) tiles

$$Q_0 P_0 Q_1 P_1 \cdots Q_n P_n \cdots$$

such that, for all  $i \geq 0$ , it holds that  $Q_i \subseteq \llbracket \beta_i \rrbracket$ ,  $P_i \subseteq \llbracket \alpha_i \rrbracket$ ,  $f(t_i) \in Q_i$ , and  $f(t) \in P_i$ , for all  $t \in (t_i, t_{i+1})$ . Consider then the following sequence of nodes:

$$\pi = (Q_0) (Q_0, P_0, Q_1) (Q_1) (Q_1, P_1, Q_2) \cdots (Q_n) (Q_n, P_n, Q_{n+1}) \cdots$$

It is immediate to observe that each such node has a non-empty denotation, as  $f$  passes through at least one point in each of them. Hence, all those nodes belong to  $V$ . Moreover, by the definition of  $E$ , all pairs  $((Q_i), (Q_i, P_i, Q_{i+1}))$  and  $((Q_i, P_i, Q_{i+1}), (Q_{i+1}))$  belong to  $E$ . Thus,  $\pi$  is a proper infinite path in  $\text{Pth}(G)$ . By the way in which we constructed  $\pi$ , it is also evident that  $\text{obs}(\pi) = \text{trc}(f, \tau)$ . The case in which  $f$  is left-open is very similar and is, therefore, omitted. ■

As reported in the third line in Table 2, the complexity profile of the present scenario is entirely analogous to that of Section 5.2, as it employs the same polyhedral abstraction.

### 6.2. Second Scenario: Non-Recurrent RTL and a Closed Flow

The peculiarity of reasoning about the non-recurrent fragment of RTL on infinite trajectories lies in the fact that such reasoning can be decomposed into a generalised reachability property, which holds for a prefix of the trajectory, followed by a safety property, which must persist on the remaining suffix. This decomposition allows us to address the first part using the techniques from the previous section, while the second is ensured by verifying that the trajectory eventually remains forever within a region whose observables satisfy the safety condition. Fortunately, the latter property can be verified locally as a property of the patches of the polyhedra associated with the observables.

To this end, we introduce an auxiliary proposition, *stay*, which holds in the patches of the observables where at least one trajectory remains indefinitely. That is, the patch is *trajectory unbounded*, or, using the terminology of [16], it is not *t*-bounded with respect to *Flow*. Formally, a polyhedron  $P$  is *t*-bounded *w.r.t.* *Flow* if, for all points  $x \in P$  and admissible trajectories  $f$  from  $x$ , there exists a time instant  $t \geq 0$  such that  $f(t) \notin P$ . By [16, Corollary 5.6], a convex polyhedron is *t*-bounded with respect to a convex closed *Flow* iff it is *bounded* with respect to *Flow*, which means that there is an admissible straight trajectory from each point of  $P$  that exits from  $P$  at some point. This last property can be efficiently tested via simple polyhedral operations as follows [16, Theorem 5.8]:

$$P \text{ is } t\text{-bounded iff } O_P \cap Flow = \emptyset,$$

where  $O_P$  denotes the characteristic cone of the convex polyhedron  $P$ , namely the closed polyhedron generated by the origin  $\mathbf{0}$  and all the rays of  $P$ . As a consequence, a patch  $P$  is not *t*-bounded if some admissible straight trajectory from some  $x \in P$  remains forever in  $P$ , hence there is a  $\hat{v} \in Flow$  such that  $x + t \cdot \hat{v} \in P$ , for all  $t \geq 0$ . Since  $P$  is convex, the direction  $\hat{v}$  corresponds to a ray of  $P$ , that is a direction of unboundedness of that polyhedron. Hence, for any  $y \in P$ , it holds that  $y + t \cdot \hat{v} \in P$ , for all  $t \geq 0$ . This ensures that if a patch  $P$  is not *t*-bounded, then, from any point in  $P$ , there is a trajectory that remains in  $P$  forever. These observations allow us to define the denotation of *stay* as follows:

$$[stay] = \bigcup_{\alpha \subseteq AP} \{P \in Patch(\llbracket \alpha \rrbracket) \mid O_P \cap Flow \neq \emptyset\}.$$

Note that if  $\mathbf{0} \in Flow$ , then the denotation of *stay* trivially coincides with that of  $\top$ .

Let  $\mathcal{P}_{stay}$  be the polyhedral system obtained from  $\mathcal{P}$  by adding the proposition *stay* with the denotation defined above. We can now establish the following theorem.<sup>6</sup>

---

<sup>6</sup>Recall that *last* is short for  $\neg X\top$ .

**Theorem 10.** *For all non-recurrent RTL formulae  $\varphi$ , the following holds true:*

$$\mathcal{P}, x \models^{\text{inf}} \varphi \text{ iff } \mathcal{P}_{\text{stay}}, x \models^{\text{fin}} \varphi \wedge \mathbf{F}(\text{stay} \wedge \text{last})$$

In order to prove the above theorem, we introduce the following two claims, whose proofs are reported in [Appendix A.4](#).

**Claim 1.** *Let  $\varphi$  be a non-recurrent RTL formula and  $\sigma: I \rightarrow 2^{AP}$  an infinite-time signal such that  $\sigma \models \varphi$ . Then, there exist a propositional formula  $\tilde{\eta}$  and a time instant  $\tilde{t} \in I$  with the following properties:*

- 1)  $\sigma_{\geq \tilde{t}} \models \mathbf{G}\tilde{\eta}$ ;
- 2) for all signals  $\sigma'$  with  $\sigma'_{\leq \tilde{t}} = \sigma_{\leq \tilde{t}}$  and  $\sigma'_{\geq \tilde{t}} \models \mathbf{G}\tilde{\eta}$ , it holds that  $\sigma' \models \varphi$ .

**Claim 2.** *Let  $\varphi$  be a non-recurrent RTL formula and  $\sigma: I \rightarrow 2^{AP}$  a finite-time signal such that  $\sigma \models \varphi$ . Then, for the unique infinite-time signal  $\sigma'$  with  $\sigma'(t) = \sigma(t)$ , for  $t \in I$ , and  $\sigma'(t) = \sigma(\sup(I))$ , otherwise, it holds that  $\sigma' \models \varphi$ .*

*Proof of Theorem 10.* We prove the two implications (**only if** and **if**, respectively).

- **[Only if direction].** Since  $\mathcal{P}, x \models^{\text{inf}} \varphi$ , there exists an infinite-time trajectory  $f \in \text{Traj}^{\text{inf}}(x)$  such that  $\sigma_f \models \varphi$ . By Item 1 of Claim 1, there is a propositional formula  $\tilde{\eta}$  and a time instant  $\tilde{t} \in I$  such that  $(\sigma_f)_{\geq \tilde{t}} \models \mathbf{G}\tilde{\eta}$ . Since  $f$  is well-behaved and defined over an unbounded time domain, there is a patch  $P \in \text{Patch}(\llbracket \alpha \rrbracket)$ , for some  $\alpha \subseteq AP$  satisfying  $\tilde{\eta}$ , such that  $f$  lies  $P$  in a diverging sequence of times, all bigger than  $\tilde{t}$ . Let  $\tilde{t} \leq t_0 < t_1 < t_2 < \dots$  be such a sequence, such that  $f(t_i) \in P$ , for all  $i$ . Now, by Lemma 1, there exists a trajectory  $\hat{f}$  such that (i)  $\hat{f}$  is equal to  $f$  up to  $t_0$  included, (ii)  $\hat{f}(t_i) = f(t_i)$ , for all  $i$ , and (iii)  $\hat{f}$  is a straight-line trajectory on all intervals  $(t_i, t_{i+1})$ . The second property ensures that  $\hat{f}$  is well-behaved, while the third one, together with the convexity of the patch  $P$ , ensures that  $\hat{f}$  stays forever inside  $P$  after  $t_0$ . Since  $\hat{f}_{\geq t_0}$  is an infinite-time trajectory completely included in  $P$ , we clearly have that  $P$  is not  $t$ -bounded and, thus, it is labelled with the atomic proposition *stay* in  $\mathcal{P}_{\text{stay}}$ . At this point, consider the finite-time trajectory  $f' \triangleq \hat{f}_{\leq t_0}$  and its signal  $\sigma' \triangleq \sigma_{f'}$ . Clearly,  $\sigma' \models \mathbf{F}(\text{stay} \wedge \text{last})$ , when interpreted in  $\mathcal{P}_{\text{stay}}$ . Moreover, when restricted to the original atomic propositions in  $\mathcal{P}$ , it holds that  $\sigma'_{\leq \tilde{t}} = (\sigma_f)_{\leq \tilde{t}}$  and  $\sigma'_{\geq \tilde{t}} \models \mathbf{G}\tilde{\eta}$ . By Item 2 of Claim 1, we thus have  $\sigma' \models \varphi$ . Hence,  $\sigma' \models \varphi \wedge \mathbf{F}(\text{stay} \wedge \text{last})$ , from which it immediately follows that  $\mathcal{P}_{\text{stay}}, x \models^{\text{fin}} \varphi \wedge \mathbf{F}(\text{stay} \wedge \text{last})$ .
- **[If direction].** Since  $\mathcal{P}_{\text{stay}}, x \models^{\text{fin}} \varphi \wedge \mathbf{F}(\text{stay} \wedge \text{last})$ , there exists a finite-time trajectory  $f \in \text{Traj}^{\text{fin}}(x)$  such that (i)  $\sigma_f \models \varphi$  and (ii) there is a non- $t$ -bounded patch  $P \in \text{Patch}(\llbracket \alpha \rrbracket)$ , for some  $\alpha \subseteq AP$ , such that  $f(\sup(I)) \in P$ . Since  $P$  is not  $t$ -bounded, there exists a left-open infinite-time trajectory  $\hat{f} \in \text{Traj}^{\text{fin}}(f(\sup(I)))$  starting at  $f(\sup(I))$  entirely contained in  $P$ . Let  $f' \in \text{Traj}^{\text{fin}}(x)$  be the infinite-time trajectory resulting from concatenation

of  $f$  and  $\widehat{f}$ , i.e.,  $f'(t) = f(t)$ , if  $t \leq \sup(I)$ , and  $f'(t) = \widehat{f}(t)$ , otherwise. Obviously, the signals  $\sigma$  and  $\sigma' = \sigma_{f'}$  satisfy the hypotheses of Claim 2. Hence,  $\sigma' \models \varphi$ , from which it immediately follows that  $\mathcal{P}, x \models^{\text{inf}} \varphi$ . ■

At this point, leveraging the result above, the model-checking problem for a non-recurrent RTL formula  $\varphi$  over the infinite-time trajectories of  $\mathcal{P}$  reduces to the corresponding problem for the formula  $\varphi \wedge \mathbf{F}(stay \wedge last)$  over the finite-time trajectories of the polyhedral system  $\mathcal{P}_{stay}$ . The latter can be then solved by applying Theorem 5 to the polyhedral abstraction introduced in Section 5.1.

The complexity of solving the present scenario follows from the fact that the original polyhedral system is extended with proposition *stay*, and the formula of interest is modified with the addition of the conjunct  $\mathbf{F}(stay \wedge last)$ . This does not increase the total number of patches of all observables (because the denotation of *stay* is composed of some of the old patches) but it can potentially double the number of states of the automaton for the formula, and consequently double the sufficient horizon  $H$ .

## 7. Maximal Semantics

Both the trajectory termination criteria “may” and “must” share the property that, in principle, they admit both finite-time and infinite-time trajectories. The model-checking problem for a formula in  $\text{RTL}_{\infty}$  when  $\gamma \in \{\text{may}, \text{must}\}$  can be decomposed into the two subproblems of checking the input formula  $\varphi$  under finite-time semantics  $\gamma$  and, separately, checking it under the infinite-time one. The set of initial points satisfying the original formula is the union of the sets of points satisfying the two subproblems. This decomposition can easily be verified sound and complete thanks to the following trivial equivalence, which holds for any formula  $\varphi$ :

$$\varphi \equiv (\varphi \wedge \mathbf{F} last) \vee (\varphi \wedge \mathbf{G} \neg last). \quad (5)$$

The disjunct  $\varphi \wedge \mathbf{F} last$  asks for a finite-time trajectory that satisfies  $\varphi$ , while  $\varphi \wedge \mathbf{G} \neg last$  seeks an infinite-time trajectory satisfying the formula. The finite-time model-checking subproblem for the semantics  $\gamma$  can, in turn, be reduced to the finite-time model checking problem discussed in Section 5. Note, however, that in the cases we are considering here not all finite-time trajectories are admissible. Intuitively, when  $\gamma = \text{may}$ , only finite-time trajectories ending in a point  $y$  on the edge of the invariant such that at least a trajectory starting from  $y$  is cut out by the invariant. Similarly, when  $\gamma = \text{must}$ , only finite-time trajectories ending in a point  $y$  on the edge of the invariant such that all admissible trajectories from  $y$  are cut out by the invariant. In order to reduce the two problems to the finite-time case of Section 5, we therefore need to restrict the set of trajectories considered in that problem to the correct one. To do this, we introduce a fresh atomic proposition *brink*, whose denotation depends on the semantics  $\gamma$  of interest and contains all and only the points  $x$  with the property discussed

above.

$$\begin{aligned} [\mathit{brink}]^{\text{may}} &\triangleq \{x \in \text{Inv} \mid \exists f \in \text{Traj}_\top(x). f \text{ immediately enters } \overline{\text{Inv}}\} \\ [\mathit{brink}]^{\text{must}} &\triangleq \{x \in \text{Inv} \mid \forall f \in \text{Traj}_\top(x). f \text{ immediately enters } \overline{\text{Inv}}\}. \end{aligned}$$

We can compute  $[\mathit{brink}]^{\text{must}}$  by subtracting from the invariant the set of all points that may remain in the invariant for a positive amount of time, namely  $\text{reach}^0(\text{Inv}, \text{Inv})$ . The latter can be computed via the following lemma.

**Lemma 9.** *For all polyhedra  $A$ , it holds:*

$$\text{reach}^0(A, A) = \bigcup_{P, Q \in \text{Patch}(A)} \text{reach}^0(P, Q).$$

*The above set is a polyhedron with at most  $|\text{Patch}(A)|^2$  patches.*

*Proof.* Let  $x \in A$ ,  $f \in \text{Traj}(x)$ , and  $T > 0$  be s.t.  $f(t) \in A$  for all  $t \in [0, T]$ . Let  $P \in \text{Patch}(A)$  be the patch of  $A$  containing  $x$ . We distinguish two cases. If  $f$  immediately leaves  $P$ , then by well-behavedness it stays in another patch  $Q \in \text{Patch}(A)$  during a time interval  $(0, T']$ , for some  $T' \leq T$ . By definition of  $\text{reach}^0$  (see Sec. 2.2), it holds  $x \in \text{reach}^0(P, Q)$ . If instead  $f$  stays in  $P$  during a time interval of the type  $[0, T']$ , it is easy to see that  $x \in \text{reach}^0(P, P)$ .

Vice-versa, let  $x$  belong to  $\text{reach}^0(P, Q)$ , for two patches  $P, Q \in \text{Patch}(A)$ . By definition,  $x \in P$  and there exists a trajectory that starts in  $x$  and spends a positive amount of time in  $Q$ , as required by the statement. ■

Now, the denotation of  $\mathit{brink}$  in the two semantics can be computed as follows:

$$[\mathit{brink}]^\gamma = \begin{cases} \text{reach}^0(\text{Inv}, \overline{\text{Inv}}), & \text{if } \gamma = \text{may}; \\ \text{Inv} \setminus \text{reach}^0(\text{Inv}, \text{Inv}), & \text{if } \gamma = \text{must}. \end{cases}$$

Observe that, by the definitions of the operator  $\text{reach}^0$  in Section 2.2, the set  $\text{reach}^0(\text{Inv}, \overline{\text{Inv}})$  contains the points of the invariant from where an admissible trajectory immediately enters  $\overline{\text{Inv}}$ . It can be computed using the distributivity property (1) using a number of symbolic operations that is linear in the number of patches of  $\overline{\text{Inv}}$ .

By Lemma 9, the set  $\text{Inv} \setminus \text{reach}^0(\text{Inv}, \text{Inv})$ , instead, contains those points  $x$  in the invariant from which no trajectory can remain in the invariant for a positive amount of time. Therefore, if the system reaches any such point  $x$ , it can no longer move.

**Lemma 10.** *The set  $[\mathit{brink}]^{\text{may}}$  (resp.,  $[\mathit{brink}]^{\text{must}}$ ) can be computed with a number of symbolic operations on polyhedra that is linear (resp., quadratic) w.r.t. the number of patches of  $\overline{\text{Inv}}$  (resp.,  $\text{Inv}$ ).*

Let  $\mathcal{P}_{\mathit{brink}}^\gamma$  be the polyhedral system obtained from  $\mathcal{P}$  by adding the proposition  $\mathit{brink}$  with the denotation defined according to the semantics  $\gamma$ . We can prove the following theorems.

**Theorem 11.** For all RTL formulae  $\varphi$  and semantics  $\gamma \in \{\text{may}, \text{must}\}$  the following holds true:

$$\mathcal{P}, x \models^\gamma \varphi \wedge \mathbf{G} \neg \text{last} \text{ iff } \mathcal{P}, x \models^{\text{inf}} \varphi$$

*Proof.* It suffices to show that, for all trajectories  $f \in \text{Traj}(x)$ , it holds that  $\sigma_f \models \mathbf{G} \neg \text{last}$  iff  $f \in \text{Traj}^{\text{inf}}(x)$ .

For one direction, assume  $f \in \text{Traj}^{\text{inf}}(x)$  and let  $t \in I_f$  be an arbitrary time instant in the domain of  $f$ . Since  $\text{sup}(I_f) = \infty$ , there exists a  $\bar{t} > t$  such that  $t' \in I_f$ , for all  $t < t' \leq \bar{t}$ . Therefore,  $(\sigma_f)_{\geq t} \models \mathbf{X} \top$ , hence  $(\sigma_f)_{\geq t} \models \neg \text{last}$ . By the arbitrariness of  $t$ , we obtain that  $\sigma_f, x \models \mathbf{G} \neg \text{last}$ .

For the other direction, assume  $\sigma_f \models \mathbf{G} \neg \text{last}$  but, by contradiction,  $f \in \text{Traj}^{\text{fin}}(x)$ . Then,  $(\sigma_f)_{\geq \text{sup}(I_f)} \models \text{last}$  (recall that, for finite trajectories,  $\text{sup}(f) \in I_f$ ), which contradicts  $\sigma_f \models \mathbf{G} \neg \text{last}$ . ■

**Theorem 12.** For all RTL formulae  $\varphi$  and semantics  $\gamma \in \{\text{may}, \text{must}\}$ , the following holds:

$$\mathcal{P}, x \models^\gamma \varphi \wedge \mathbf{F} \text{last} \text{ iff } \mathcal{P}_{\text{brink}}^\gamma, x \models^{\text{fin}} \varphi \wedge \mathbf{F}(\text{brink} \wedge \text{last})$$

*Proof.* It suffices to show that, for all  $f \in \text{Traj}^\gamma(x)$ ,  $\sigma_f \models \mathbf{F} \text{last}$  iff  $f \in \text{Traj}_{\text{brink}}^{\text{fin}}(x)$  and  $\sigma_f \models \mathbf{F}(\text{brink} \wedge \text{last})$ .

Assume  $\sigma_f \models \mathbf{F} \text{last}$ . Since  $\mathbf{F} \text{last} \equiv \neg \mathbf{G} \neg \text{last}$  and by Theorem 11, we have that  $f \in \text{Traj}^{\text{fin}}(x)$ , which implies  $f \in \text{Traj}_{\text{brink}}^{\text{fin}}(x)$ . Let  $y \triangleq f(\text{sup}(I_f))$ . Clearly  $\mathcal{P}, y \models^{\text{fin}} \text{last}$ . By the definition of  $\text{Traj}^\gamma(x)$ , if  $\gamma = \text{may}$ , then there exists  $f' \in \text{Traj}_\top(y)$  such that  $f'$  immediately enters  $\overline{\text{Inv}}$ . If, on the other hand,  $\gamma = \text{must}$ , then for all  $f' \in \text{Traj}_\top(y)$ , it holds that  $f'$  immediately enters  $\overline{\text{Inv}}$ . In either case,  $f(y) = f'(y) \in [\text{brink}]^\gamma$ . Hence,  $\sigma_f \models \mathbf{F}(\text{brink} \wedge \text{last})$  as desired.

For the other direction, assume  $f \in \text{Traj}_{\text{brink}}^{\text{fin}}(x)$ ,  $\sigma_f \models \mathbf{F}(\text{brink} \wedge \text{last})$ , and let  $y \triangleq f(\text{sup}(I_f))$ . Then,  $(\sigma_f)_{\geq \text{sup}(I_f)} \models \text{last}$  and  $y \in [\text{brink}]^\gamma$ . If  $\gamma = \text{must}$ , then, for all  $f' \in \text{Traj}_\top(y)$ ,  $f'$  immediately enters  $\overline{\text{Inv}}$ , hence  $f$  is a finite-time trajectory contained in  $\text{Traj}^{\text{must}}(x)$ . Similarly, if  $\gamma = \text{may}$ , then, some  $f' \in \text{Traj}_\top(y)$  immediately enters  $\overline{\text{Inv}}$ , hence  $f$  is a finite-time trajectory contained in  $\text{Traj}^{\text{must}}(x)$ . We, thus, conclude that  $f \in \text{Traj}^\gamma(x)$  for  $\gamma \in \{\text{may}, \text{must}\}$ . Since, in addition,  $\sigma_f \models \mathbf{F} \text{last}$ , we obtain the conclusion. ■

The following corollary, which is an immediate consequence of the previous two lemmata, proves soundness and completeness of the decomposition.

**Corollary 2.** For all RTL formulae  $\varphi$  and semantics  $\gamma \in \{\text{may}, \text{must}\}$ , the following holds:

$$\mathcal{P}, x \models^\gamma \varphi \text{ iff } [\mathcal{P}_{\text{brink}}^\gamma, x \models^{\text{fin}} \varphi \wedge \mathbf{F}(\text{brink} \wedge \text{last}) \text{ or } \mathcal{P}, x \models^{\text{inf}} \varphi]$$

### 7.1. First Scenario: Omnidirectional Flow

According to Corollary 2, the model-checking problem for  $\varphi$  can be decomposed into the corresponding problem for the formula  $\varphi \wedge \mathbf{F}(brink \wedge last)$  over the finite-time trajectories of the polyhedral system  $\mathcal{P}_{brink}^\gamma$  and the problem of verifying  $\varphi$  over the infinite-time trajectories of  $\mathcal{P}$ . The first problem, in turn, can be solved by applying Theorem 5 on the polyhedral abstraction of Section 5.2. The second one, instead, uses the polyhedral abstraction of 6.1 and, again, applies Theorem 5.

The resulting complexity (see Table 2) is dominated by the size of the polyhedral abstraction for the finite-time case, where the number of patches  $B$  of *brink* enters the picture.<sup>7</sup> This leads to an upper bound of  $(BK)^3 + BK$  nodes in the polyhedral abstraction (where  $K$  is the number of patches of the original observables).

Observe that we can avoid solving the subproblem on finite-time trajectories when  $\gamma = \text{must}$ , as it becomes trivial and always has a negative answer. Indeed, the set of trajectories  $\text{Traj}^{\text{must}}$  coincides with  $\text{Traj}^{\text{inf}}$ , since any point  $x$  in *Inv* admits an infinite-time trajectory that stays in  $x$  forever. Hence, in this case, only the subproblem on infinite-time trajectories needs to be solved.

### 7.2. Second Scenario: Forced Motion and Bounded Invariant

In this scenario, we assume that the origin lies outside the flow constraint ( $\mathbf{0} \notin \text{cl}(\text{Flow})$ ) and that the invariant is bounded. We call the first condition *forced motion* because the system cannot stay still, and cannot even slow down arbitrarily. Under these assumptions, no infinite-time trajectories exist.

**Lemma 11.** *If  $\mathbf{0} \notin \text{cl}(\text{Flow})$  and *Inv* is bounded, for all  $x \in \mathbb{R}^n$  it holds that  $\text{Traj}^{\text{inf}}(x)$  is empty.*

*Proof.* We compute an upper bound for the time any trajectory may spend within the invariant. Consider the distance between the origin  $\mathbf{0}$  and the (closure of) the flow. Let  $\vec{d}$  be the distance vector between the origin and *Flow*, and  $\hat{d}$  its versor. Every point in *Flow* has a projection on  $\hat{d}$  whose length is at least  $|\vec{d}|$ . Moreover, since the invariant is bounded, its scalar projection on (the line of)  $\hat{d}$  is also bounded, with supremum projection  $M$ .

Given an arbitrary trajectory  $f$ , we consider its scalar projection on the line of  $\hat{d}$  as a function  $g : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}$ . For its time derivative, the following holds:

$$\dot{g}(t) = \hat{d} \cdot \dot{f}(t) \geq |\vec{d}| > 0.$$

Hence, after time  $\frac{M}{|\vec{d}|}$  the trajectory  $f$  will have a scalar projection on  $\hat{d}$  greater than any point in *Inv*, which implies that  $f$  has exited the invariant. ■

As a consequence of the above lemma and Corollary 2, we obtain the following.

---

<sup>7</sup>Note that  $B$  is strictly related to the shape of the invariant, because in both may and must semantics the denotation of *brink* is a subset of the boundary of the invariant.

**Corollary 3.** For all RTL formulae  $\varphi$  and semantics  $\gamma \in \{\text{may}, \text{must}\}$ , if  $0 \notin \text{cl}(\text{Flow})$  and  $\text{Inv}$  is bounded, then the following holds:

$$\mathcal{P}, x \models^\gamma \varphi \text{ iff } \mathcal{P}_{\text{brink}}^\gamma, x \models^{\text{fin}} \varphi \wedge \mathbf{F}(\text{brink} \wedge \text{last})$$

At this point, the model-checking problem can be solved as in Section 5.1. As discussed in Section 7.1, the addition of *brink* increases the overall number of patches from  $K$  to at most  $BK$ , and the maximum number of patches of an observable from  $K_{\max}$  to at most  $BK_{\max}$ , where  $B$  is the number of patches of  $[\text{brink}]^\gamma$ . Moreover, the conjunct  $\mathbf{F}(\text{brink} \wedge \text{last})$  added to the formula doubles the sufficient horizon, leading to the complexity reported in Table 2.

### 7.3. Third Scenario: Non-Recurrent RTL and a Closed Flow

Similarly to the First Scenario (Sec. 7.1), the model-checking problem for  $\varphi$  is decomposed into the corresponding problem for the formula  $\varphi \wedge \mathbf{F}(\text{brink} \wedge \text{last})$  over the finite-time trajectories of the polyhedral system  $\mathcal{P}_{\text{brink}}^\gamma$  and the problem of verifying  $\varphi$  over the infinite-time trajectories of  $\mathcal{P}$ . The first problem is again solved through Theorem 5 applied to the polyhedral abstraction of Section 5.1. The second one, instead, is addressed in Section 6.2 and, in turn, also reduces to the model checking of finite-time trajectories in the polyhedral system  $\mathcal{P}_{\text{stay}}$ .

The last block in Table 2 reports the complexity profile of this scenario. The block is divided into two lines, that refer to the above decomposition into two sub-problems, except for the last cell that contains the overall complexity. The latter is dominated by the first problem, i.e., analysing  $\varphi \wedge \mathbf{F}(\text{brink} \wedge \text{last})$  over the polyhedral system  $\mathcal{P}_{\text{brink}}^\gamma$ .

## 8. An On-The-Fly Algorithm for the Finite-Time Semantics

In this section, we describe the on-the-fly algorithm that solves the existential denotation problem for  $\text{RTL}_f$  on polyhedral systems, that is RTL under the finite semantics considered in Section 5. The algorithm solves the problem without computing the polyhedral abstraction in advance. The basic idea is that it explores backwards each path of the automaton for the discretised formula  $\widehat{\varphi}^{\text{fin}}$  starting from an accepting state and, while doing so, it computes incrementally the denotation of the node of the polyhedral abstraction corresponding to the that path.

Unless differently specified, we consider a fixed  $\text{RTL}_f$  formula  $\varphi$  over the set of atomic propositions  $AP$ , and a fixed polyhedral system  $\mathcal{P}$  on  $AP$ . Before describing the algorithm itself, we introduce two auxiliary operators on polyhedra.

Type	Name	Role	Symbol
$\{0, 1, \dots, k\} \rightarrow S$	Discrete run	Behaviour of a finite automaton	$r^{\text{d}}$
$I \rightarrow S$	Continuous run	Continuous behaviour of a finite automaton	$r^{\text{c}}$
$I \rightarrow (\mathbb{R}^n \times S)$	Hybrid run	Pairing of a trajectory and a continuous run	$\rho$

Table 3: Notation used in Section 8: three types of runs of an automaton.

### 8.1. The Finite Automaton

Our algorithm works on a finite automaton that checks the satisfaction of  $\varphi$ , while ensuring a number of extra properties. The automaton is obtained by applying the classic LTL<sub>f</sub>-to-automata construction to the formula  $\widehat{\varphi}^{\text{fin}}$  defined by equation (4).

Let  $\mathcal{A}_\varphi^{\text{fin}} = (2^{\widehat{AP}}, S, \delta, \lambda, S_0, S_F)$  be the finite automaton corresponding to  $\widehat{\varphi}^{\text{fin}}$ , according to Theorem 1. We assume w.l.o.g. that  $\mathcal{A}_\varphi^{\text{fin}}$  satisfies the following properties. Properties (a) and (b) are directly encoded in  $\widehat{\varphi}^{\text{fin}}$ , while property (c) can easily be enforced via a simple modification of the automaton.

**Proposition 1.** *The finite automaton  $\mathcal{A}_\varphi^{\text{fin}}$  satisfies the following properties:*

- (a) *the underlying graph is bipartite in  $(S_{\text{sing}}, S_{\text{open}})$ , where  $S_{\text{sing}}$  is the set of all states labelled with *sing*, while  $S_{\text{open}}$  is its complement;*
- (b) *all the final states are labelled with the proposition *sing*;*
- (c) *the initial states have no predecessors.*

We denote by  $\text{Run}^{\text{d}}(f)$  the set of all initial runs of  $\mathcal{A}_\varphi^{\text{fin}}$  on the discrete traces of  $f$ . For a trajectory  $f$  and a time slicing  $\tau = \{t_i\}_{i=0}^k \in TS(\sigma_f)$ , let  $w$  be the corresponding discrete trace and  $r^{\text{d}}$  one of the runs of  $\mathcal{A}_\varphi^{\text{fin}}$  on  $w$ . We define the *continuous run*  $r^{\text{c}}$  for  $r^{\text{d}}$  and  $\tau$  as follows:

$$r^{\text{c}}(t) = \begin{cases} r^{\text{d}}(2 \cdot i) & \text{if } t = t_i, \text{ for some } i, \\ r^{\text{d}}(2 \cdot i + 1) & \text{if } t \in (t_i, t_{i+1}), \text{ for some } i. \end{cases}$$

We denote with  $\text{Run}^{\text{c}}(f)$  the set of continuous runs induced by  $f$  as just described.

Moreover, we define the notion of *hybrid run* as the function  $\rho = \lambda t. (f(t), r^{\text{c}}(t))$  pairing a trajectory with one of its continuous runs. Let  $\text{HRun}(x)$  be the set of hybrid runs  $(f, r^{\text{c}})$ , where  $f \in \text{Traj}_{\text{wb}}(x)$  and  $r^{\text{c}} \in \text{Run}^{\text{c}}(f)$ .

### 8.2. The On-the-fly Algorithm

The main steps in the procedure to solve the existential denotation problem are illustrated by pseudo-code of the function  $\exists\text{Denot}(\cdot)$  in Algorithm 1. Theorem 1 at the end of this section describes the top-level invocations that start the process, which begins from a final state of the automaton and then works recursively backward towards the initial states.

Roughly speaking, a call to  $\exists\text{Denot}(s, P, X, V)$  computes the points from where there exists a hybrid run of the automaton ending in the state  $s$  and in a point in the convex polyhedron  $X$ . Moreover,  $X$  is assumed to be contained in  $P$ , and  $P$  must be a patch of  $\llbracket s \rrbracket$ . The role of the parameter  $V$  is explained below. In the following, for a state  $s \in S$ , let  $\text{type}(s) = 0$ , if  $\text{sing} \in \lambda(s)$ , and  $\text{type}(s) = +$ , otherwise.

To ensure termination, the algorithm keeps track of the patches associated with *open states* in  $S_{\text{open}}$  that have been visited in the current sequence of recursive calls. Those are the patches in which the induced trajectory must

---

**Algorithm 1:** Function  $\exists\text{Denot}(s, P, X, V)$ . For simplicity, we omit from the notation two implicit arguments: the finite automaton  $\mathcal{A}_\varphi^{\text{fin}} = (2^{\widehat{A^P}}, S, \delta, \lambda, S_0, S_F)$  and the polyhedral system  $\mathcal{P}$ .

---

**input** :  $s \in S$ ;  
            $P$ : convex polyhedron in  $\text{Patch}(\llbracket s \rrbracket)$ ;  
            $X$ : convex polyhedron included in  $P$ ;  
            $V$ : map from states  $u \in S$  to a subset of the patches of  $\llbracket u \rrbracket$ ;  
**output** : A polyhedron in  $\mathbb{R}^n$

```

1 if  $s \in S_0$  then return  $\text{reach}^{\text{type}(s)}(X, \mathbb{R}^n)$ 
2 Result  $\leftarrow \emptyset$ 
3  $V' \leftarrow$  if  $s \in S_{\text{sing}}$  then  $V$  else  $V[s \mapsto V(s) \cup \{P\}]$ 
4 foreach state  $s' \in S$  such that  $(s', s) \in \delta$  do
5    $A \leftarrow \llbracket s' \rrbracket \setminus V(s')$ 
6    $A' \leftarrow \text{reach}^{\text{type}(s')}(A, X)$ 
7    $\{(Q_1, Y_1), \dots, (Q_n, Y_n)\} \leftarrow \text{split}(A', A)$ 
8   for  $i = 1, \dots, n$  do
9      $\llbracket \text{Result} \leftarrow \text{Result} \cup \exists\text{Denot}(s', Q_i, Y_i, V')$ 
10 return Result

```

---

spend some positive amount of time. This information is kept in the map  $V$ , that associates with each state  $s$  the set of patches of  $\llbracket s \rrbracket$  already encountered by the algorithm.

Line 1 takes care of the base case of the algorithm, when  $s$  is an initial state. In this case, if  $s$  belongs to  $S_0$ , then it returns  $X$  itself (observe that  $X = \text{reach}^0(X, \mathbb{R}^n)$ ). If, however,  $s \in S_{\text{open}}$ , then the correct answer contains the points in  $X$  from where a trajectory can spend some time in  $X$ . These points are precisely those contained in  $\text{reach}^+(X, \mathbb{R}^n)$ . If  $s$  is not initial, an updated map  $V'$  is computed, where the patch  $P$  is added to  $V(s)$  if  $s$  is an open state (Line 3). The for loop at Lines 4–9 iterates over the incoming edges of  $s$ . For each such edge  $(s', s)$ , Line 5 sets  $A$  to the region of  $\llbracket s' \rrbracket$  that has *not* been already visited. Line 6 computes the set of points of  $A$  that can reach some point in  $X$ , either leaving  $A$  immediately, if  $s'$  is a singular state ( $\text{type}(s) = 0$ ), or lingering in  $A$  for some time, if it is open ( $\text{type}(s) = +$ ). Line 7 splits the resulting set  $A'$  into a set of distinct pairs  $(Q_i, Y_i)$ , where  $Y_i$  is the maximal convex polyhedron contained in  $A'$  and in the patch  $Q_i$  of  $A$ . Each such pair  $(Q_i, Y_i)$ , then, gives rise to a recursive call on the state  $s'$  with targets  $Y_i$  and  $Q_i$  at Line 9. The results of all such calls are gathered in **Result**, which is returned at Line 10.

The following lemmata state the characteristic properties of the function  $\exists\text{Denot}$ , namely termination (Lemma 12), and soundness and completeness (Lemma 14).

**Lemma 12.** *For all convex polyhedra  $P, X$  such that  $P \in \text{Patch}(\llbracket s \rrbracket)$  and  $X \subseteq P$ , and maps  $V : S \rightarrow_s 2^{\text{Patch}(\llbracket s \rrbracket)}$ , the call to  $\exists\text{Denot}(s, P, X, V)$  terminates after at*

most  $m^{O(m^2 \cdot |S|)}$  symbolic operations on polyhedra, with  $m$  the maximum number of patches in the denotation of any state.

*Proof.* First, we prove that the recursion depth is bounded by  $2 \cdot \sum_{s \in S} |\text{Patch}(\llbracket s \rrbracket)|$ . Let  $\chi = (s_0, P_0, X_0, V_0), (s_1, P_1, X_1, V_1), \dots, (s_j, P_j, X_j, V_j), \dots, (s_k, P_k, X_k, V_k), \dots$  be the sequence of arguments in a stack of recursive calls to  $\exists\text{Denot}$ , with  $(s_0, P_0, X_0, V_0)$  being the bottom of the stack. Recall that by design  $s_j \in S$  and  $P_j$  is one of the patches of  $\llbracket s_j \rrbracket$ . Moreover, observe that  $V_j \subseteq V_k$ , for every  $k > j$ . If  $s_j \in S_{\text{open}}$ , Line 3 ensures that, in every recursive call issued at recursion level  $j$  (Line 9), the patch  $P_j$  is inserted in  $V'(s_j)$ . From that point on, *i.e.*, at recursion levels  $k > j$ , if state  $s_j$  is considered at Line 4 as value for  $s'$ , Line 5 ensures that the first argument of *reach* at Line 6 does not contain Patch  $P_j$ . Therefore,  $P_j$  is not passed to the next recursive call as a possible value for one of the  $Q_i$ 's. Hence, either  $s_k \neq s_j$  or  $P_k \neq P_j$ . Equivalently, the pair  $(s_j, P_j)$  cannot occur again in the sequence  $\chi$ . It follows that the sequence  $\chi$  contains no duplicate pairs whose state belongs to  $S_{\text{open}}$ , which proves the bound on the recursion depth. Termination follows from the fact that the number of recursive calls at each level is plainly finite. As to the bound on the symbolic operations, observe that, since the output of *reach*<sup>+</sup> contains at most  $m^{O(m)}$  patches and the loop at Line 4 iterates on the states of the automaton, the branching degree of the recursion tree of the algorithm is bounded by  $|S| \cdot m^{O(m)}$ . Its depth, instead, is bounded by  $2|S|m$ , as shown above. Hence, the overall number of symbolic operations required by algorithm is bounded by  $m^{O(m^2 \cdot |S|)}$ . ■

A hybrid run  $\rho$ , with time slicing  $\{t_i\}_{i=0}^k$ , ends in the pair  $(X, s)$ , for a set of points  $X \subseteq \mathbb{R}^n$  and a state  $s \in S$ , if either  $s \in S_{\text{sing}}$  and  $\rho$  is in  $(X, s)$  at the last instant of time in its domain, or  $s \in S_{\text{open}}$  and  $\rho$  resides in  $(X, s)$  for some open time interval ending in  $t_k$ . Formally:

- if  $s \in S_{\text{sing}}$ , then  $\rho(t_k) \in X \times \{s\}$ ;
- otherwise, there exists  $t^* \in (t_{k-1}, t_k)$  such that  $\rho(t) \in X \times \{s\}$ , for all  $t \in [t^*, t_k)$ .

Moreover, we denote by  $\text{Visited}(\rho)$  the set of pairs  $(P, s)$ , composed of a patch  $P \in \text{Patch}(\llbracket s \rrbracket)$  and a state  $s$ , traversed by  $\rho$  at any time. We say that a hybrid run  $\rho$  *avoids* a pair  $(P, s)$  if  $(P, s) \notin \text{Visited}(\rho)$ . This notion of avoidance generalises to pairs  $(A, s)$ , with  $A$  a set of patches, and to sets of such pairs, in the obvious way.

The following lemma shows that for every hybrid run  $\rho$  there exists a similar hybrid run  $\rho'$  that crosses a given pair  $(P, s)$ , with  $s \in S_{\text{open}}$ , at most once. This is instrumental in proving the completeness of Algorithm 1 in Lemma 14.

**Lemma 13.** *For all hybrid runs  $\rho$ , states  $s \in S_{\text{open}}$ , and patches  $P \in \text{Patch}(\llbracket s \rrbracket)$ , there exists a hybrid run  $\rho'$  such that:*

- $\rho'$  starts and ends in the same pairs as  $\rho$ ;
- $\rho'$  passes at most once through the pair  $(P, s)$ ;

- $Visited(\rho') \subseteq Visited(\rho)$ ;
- the length of the shortest discrete trace of  $\rho'$  is smaller than or equal to that of  $\rho$ .

Correctness of the algorithm is established by the following result.

**Lemma 14.** *For all states  $s \in S$ , convex polyhedra  $P \in Patch(\llbracket s \rrbracket)$  and  $X \subseteq P$ , and maps  $V : S \rightarrow_s 2^{Patch(\llbracket s \rrbracket)}$  such that  $P \notin V(s)$ , we have that  $\exists Denot(s, P, X, V)$  returns the set of all points  $x$  from which there is a hybrid run  $\rho \in HRun(x)$  such that: (a)  $\rho$  ends in  $(X, s)$ ; (b)  $\rho$  avoids  $V$ ; (c) if  $s \in S_{open}$ , then  $\rho$  avoids  $(P, s)$ , except for the last slice.*

The following theorem, whose proof can be found in Appendix [Appendix A.5](#), describes the initial arguments required by Algorithm 1 to solve the existential denotation problem.

**Theorem 1.** *For all  $RTL_f$  formulas  $\varphi$  and polyhedral systems  $\mathcal{P}$  on the same set of atomic propositions, let  $\widehat{\varphi}^{fin}$  be the corresponding  $LTL_f$  formula,  $\mathcal{A}_\varphi^{fin}$  be the finite automaton for  $\widehat{\varphi}^{fin}$ , and let*

$$X = \bigcup_{s \in S_F} \bigcup_{P \in Patch(\llbracket s \rrbracket)} \exists Denot(s, P, P, \emptyset).$$

*Then,  $X$  is the set of points from which there exists a trajectory that satisfies  $\varphi$ .*

*Proof.* Assume there exists a trajectory  $f$  from point  $x$  that satisfies  $\varphi$ , i.e.,  $x = f(0)$  and  $\sigma_f \models \varphi$ . Let us pick an arbitrary  $\tau = \{t_i\}_{0 \leq i \leq k}$  in  $TS(\sigma_f)$  and let  $y \triangleq f(t_k)$ . Then, by Theorem 3,  $trc(\sigma_f, \tau) \models \widehat{\varphi}^{fin}$ . By definition of  $\mathcal{A}_\varphi^{fin}$ , there exists an accepting run  $r \in Runs(\mathcal{A}_\varphi^{fin})$  for  $trc(\sigma_f, \tau)$  that ends in some final state  $s \in S_F$ . Let  $\alpha$  be the last symbol of  $trc(\sigma_f, \tau)$ , then  $y$  belongs to some patch  $P$  of  $\llbracket s_F \rrbracket = \llbracket \alpha \rrbracket$ . Let now  $\rho = (f, r^c)$  be the hybrid run from  $x$  whose second component  $r^c$  is the continuous run of  $r$  and  $\tau$ . Clearly,  $\rho$  ends in  $(P, s_F)$ , hence it satisfies condition (a) of the statement of Lemma 14 (conditions (b) and (c) hold trivially for  $\rho$ ). Therefore, Lemma 14 ensures that  $x \in \exists Denot(s, P, P, \emptyset)$  and the thesis follows.

For the other direction, let  $x$  be a point in  $\exists Denot(s, P, P, \emptyset)$ , for some  $s \in S_F$  and  $P \in Patch(\llbracket s \rrbracket)$ . By Lemma 14, there exists a hybrid run  $\rho = (f, r^c)$  from  $x$  that ends in  $(P, s)$ , where  $P$  is a patch of  $\llbracket s \rrbracket$  and  $r^c$  is a continuous run of some discrete run  $r \in Runs(\mathcal{A}_\varphi^{fin})$  and some time-splitting  $\tau$  for  $f$ . The run  $r$  is accepting since it ends in the same final state  $s \in S_F$  as  $r^c$  and it accepts the word  $trc(\sigma_f, \tau)$ . This means that  $trc(\sigma_f, \tau) \models \widehat{\varphi}^{fin}$  and Theorem 3, then, ensures that  $\sigma_f \models \varphi$ . ■

## 9. Experiments

In this section, we report on the experiments performed with our implementation, which is based on Parma Polyhedral Library [8] as the underlying engine

for the symbolic manipulation of polyhedra. As shown in Figure 8, our prototype implementation starts from an  $\text{RTL}_f$  formula  $\varphi$  and computes its discretisation  $\hat{\varphi}^{\text{fin}}$  according to Equation (4). This formula is translated into standard LTL (see [27]) in order to obtain a non-deterministic Büchi automaton recognising its models using SPOT [28]. The NBA is then turned into an NFA recognising the finite traces satisfying  $\hat{\varphi}^{\text{fin}}$ . The obtained automaton, together with the polyhedral system providing the flow constraints and the polyhedral denotations of the atomic propositions of  $\varphi$ , are finally fed to  $\exists\text{Denot}$  (Algorithm 1).

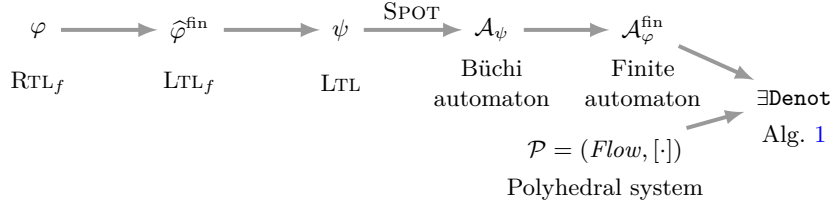


Figure 8: Architecture of the prototype implementation.

We ran some experiments based on the two-tank model described in the introduction. The experiments consist of two families of  $\text{RTL}_f$  properties, called  $\varphi_k^{\text{gap}}$  and  $\varphi_k^{\text{nogap}}$ , of the following form:

$$\varphi_k^* \triangleq \mathbf{G} \text{inv} \wedge t_0 \wedge \mathbf{G} t_{max} \wedge \underbrace{\mathbf{F} (p \wedge \mathbf{F} (q \wedge \dots \wedge \mathbf{F} (p \wedge \mathbf{F} q)))}_{k \text{ times}}$$

where  $k \geq 1$ ,  $\star \in \{\text{gap}, \text{nogap}\}$ , and the interpretations of the atomic propositions is reported in Table 4.

	$[p]$	$[q]$	$[\text{inv}]$	$[t_0]$	$[t_{max}]$
$\varphi_k^{\text{gap}}$	$a \geq b + 1$	$b \geq a + 1$	$a \geq 0 \wedge b \geq 0$	$t = 0$	$t \leq 10$
$\varphi_k^{\text{nogap}}$	$a > b$	$b > a$	$a \geq 0 \wedge b \geq 0$	$t = 0$	$t \leq 10$

Table 4: Interpretations of the atomic propositions for the two families of  $\text{RTL}_f$  formulae.

Both families require a trajectory that satisfies the invariant  $\text{inv}$ , starts at time  $t = 0$  (represented by the proposition  $t_0$ ) and ends at time 10 (enforced by the formula  $\mathbf{G} t_{max}$ ), and alternates  $k$  times between the propositions  $p$  and  $q$ . The only difference between the two families is in the polyhedral interpretations  $[p]$  and  $[q]$  of the atomic propositions  $p$  and  $q$ .

From a semantic standpoint, the first family  $\varphi_k^{\text{gap}}$  requires a trajectory to alternate  $k$  times between two disjunct and non-adjacent half-spaces. Since the flow constraint is a bounded (convex) polyhedron, the intensities of the derivatives are bounded, hence there is a minimum amount of time that any

trajectory, reaching a point of the half-space  $a \geq b + 1$ , requires to reach the half-space  $b \geq a + 1$ . As a consequence, the number of alternations possible from different points may differ. The further away from the border of the half-spaces a point is, the fewer alternations are possible.

In the second family  $\varphi_k^{\text{nogap}}$ , instead, the two half-spaces between which to alternate are adjacent. Therefore, no minimum time is needed to move from one to the other. This means that, if a trajectory can reach  $a > b$  and, from there, also reach  $b > a$ , then it may keep alternating between the two an arbitrary number of times.

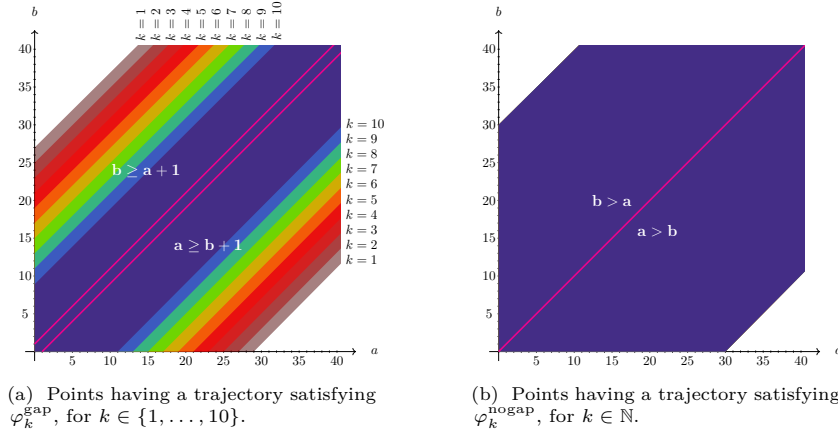


Figure 9: The results of the experiments for the two families of  $\text{RTL}_f$  properties.

Figure 9 shows the denotations of the two families of formulas, both limited to the cross section for  $t = 0$ . In particular, Figure 9a shows the different regions of points satisfying the  $\text{RTL}_f$  property indexed with the corresponding value of  $k$ . As explained above, the bigger the value of  $k$ , the smaller the region of points. For example, only the points in the dark blue region in the middle satisfy  $\varphi_{10}^{\text{gap}}$ , whereas the points satisfying  $\varphi_9^{\text{gap}}$  additionally include the two light blue strips. Observe also that the region of points in the half-space  $a \geq b + 1$  satisfying the property  $\varphi_k^{\text{gap}}$  is bigger than the region of points in the half-space  $b \geq a + 1$  that satisfies the same property. This is due to the fact that a trajectory from the points in latter region must spend additional time to first reach the half-space  $a \geq b + 1$ , leaving less time, with respect to the points in the former region, to perform the alternations. Figure 9b, instead, is perfectly symmetric and shows that all the points from where one can reach the diagonal  $a = b$  in the allotted time can alternate between the two half-spaces  $k$  times, regardless of the value of  $k$ .

We refer to [29], for a more comprehensive and in-depth analysis of the experimental results and to [30] for an implementation of the proposed algorithms and associated benchmarks.

## 10. Conclusions

In this paper, we have addressed the problem of model checking RTL on polyhedral systems governed by differential inclusions. By leveraging a symbolic algorithm that computes existential denotations, we provided an exact approach to determining the set of initial states from which a given temporal formula holds. Our methodology relies on a translation from RTL to classical LTL, followed by automata-based verification techniques combined with polyhedral abstractions.

*Choice of semantics.* We studied decidability of model-checking problem for the logic under four different semantics, which capture different forms of termination conditions for the system trajectories, namely finite-time, infinite-time, may-exit and must-exit. Such semantics exhibit different computational properties and may be employed in different concrete modelling scenarios. From a practical perspective, indeed, the choice of semantics should be guided by the modelling intent. The two crucial factors are the kind of temporal property of interest and the origin and meaning of the model invariant. The finite-time semantics is the natural option for properties witnessed by a finite execution prefix, such as reachability or eventual task completion, whereas the infinite-time semantics is more appropriate for genuinely ongoing behaviours, including liveness, recurrence, fairness, and surveillance. Concerning the invariant and the boundary behaviour, the may-exit semantics is suitable when leaving the current invariant corresponds to a possible transfer to another mode of a larger hybrid system, while the must-exit semantics is preferable when the invariant represents a hard physical constraint and executions should remain inside it unless every admissible evolution leaves it.

The results demonstrate that the model-checking problem for RTL is decidable in general for finite-time trajectories, while for the infinite-time ones it requires specific constraints on the system dynamics or on the considered formulae. We further introduced an on-the-fly symbolic algorithm that avoids the explicit product construction between the automaton of the formula and the polyhedral abstraction, improving space efficiency and practical performance. Experimental results from our prototype implementation validate the approach.

Future directions include extending our framework to richer temporal properties, possibly including metric operators [10, 11], optimising computational efficiency, and exploring further decidability boundaries for infinite-time trajectories.

The results of this work have potential applications in the formal verification of cyber-physical systems, in robotics, and in control theory. To this aim, we briefly discuss a potential extension to multi-mode hybrid systems.

*Generalisation to multi-location systems.* Most results developed in this paper can be extended to hybrid systems with multiple locations (aka modes), specifically Linear Hybrid Automata [20] (LHAs), provided that one sets an upper bound to the number of location switches allowed. Such systems contain a finite set of discrete states, called locations, each characterised by a different

flow constraint  $Flow$ . One can naturally generalise the semantics of RTL to trajectories that span multiple locations and analyse the existential denotation problem in this framework.

For instance, we outline the generalisation for the finite-time semantics, using the on-the-fly approach from Section 8.2. To this end, we extend the algorithm  $\exists\text{Denot}$  with two extra parameters: for a location  $l$  and a state  $s$  from  $\mathcal{A}_\varphi^{\text{fin}}$ ,  $\exists\text{Denot}_{l,s}(s', P, Q, \emptyset)$  represents an invocation to  $\exists\text{Denot}(s', P, Q, \emptyset)$  where the flow constraint used by the two *reach* operators is taken from location  $l$ , and  $s$  is treated as the only initial state of the automaton, i.e.,  $S_0 = \{s\}$ . With this convention, the multi-location problem with at most  $k$  location switches can be solved by the following recursion:

$$M(l, k, s) \triangleq \begin{cases} \bigcup_{\substack{s_F \in S_F \\ P_F \in \text{Patch}(\llbracket s_F \rrbracket)}} \exists\text{Denot}_{l,s}(s_F, P_F, P_F, \emptyset), & \text{if } k = 0; \\ \bigcup_{\substack{l \rightarrow l', s' \in S, \\ P' \in \text{Patch}(\llbracket s' \rrbracket)}} \exists\text{Denot}_{l,s}(s', P', P' \cap M(l', k-1, s'), \emptyset), & \text{otherwise.} \end{cases}$$

$M(l, k, s)$  contains the points of the denotation of  $s$  that, starting in location  $l$ , can reach an accepting state after exactly  $k$  location switches.

*A multi-location example.* As a simple example of such an extension, consider an autonomous robot moving in a planar workspace, whose continuous state is  $(x, y, b)$ , where  $(x, y)$  denotes the robot position and  $b$  its battery energy content. The discrete locations may encode directional motion modes, such as North, South, East, and West, together with a dedicated location Recharge, enabled only inside a physical region labelled *dock*, in which the battery can be replenished. A nested reachability property such as

$$\text{charged} \cup \left( \text{target} \vee \left( \text{dock} \cup \left( \text{charged} \cup \text{target} \right) \right) \right)$$

expresses that the robot must reach a target region while maintaining its battery above a minimum charge level (proposition *charged*), possibly by spending some time in the recharging area (proposition *dock*). This illustrates how multi-location hybrid models naturally capture motion-planning tasks in which continuous motion, discrete controller switches, and resource constraints must be handled together.

## References

- [1] R. Poovendran, Cyber-physical systems: Close encounters between two parallel worlds, Proceedings of the IEEE 98 (8) (2010) 1363–1366.
- [2] T. Henzinger, The theory of hybrid automata, in: 11th IEEE Symp. Logic in Comp. Sci., 1996, pp. 278–292. [path\(doi : 0.1109/LICS.1996.561342\)](https://doi.org/10.1109/LICS.1996.561342).

- [3] T. Henzinger, P. Kopke, A. Puri, P. Varaiya, What’s decidable about hybrid automata?, *J. of Computer and System Sciences* 57 (1) (1998) 94 – 124. [path\(doi : 10.1006/jcss.1998.1581\)](https://doi.org/10.1006/jcss.1998.1581).
- [4] M. Reynolds, The complexity of temporal logic over the reals, *Annals of Pure and Applied Logic* 161 (8) (2010) 1063–1096.
- [5] M. Vardi, P. Wolper, Automata-Theoretic Techniques for Modal Logics of Programs, *Journal of Computer and System Sciences* 32 (2) (1986) 183–221.
- [6] G. D. Giacomo, M. Vardi, Linear Temporal Logic and Linear Dynamic Logic on Finite Traces, in: *International Joint Conference on Artificial Intelligence’13, International Joint Conference on Artificial Intelligence’ & Association for the Advancement of Artificial Intelligence Press*, 2013, pp. 854–860.
- [7] A. Duret-Lutz, E. Renault, M. Colange, F. Renkin, A. G. Aisse, P. Schlehuber-Caissier, T. Medioni, A. Martin, J. Dubois, C. Gillard, H. Lauko, From Spot 2.0 to Spot 2.10: What’s new?, in: *Proceedings of the 34th International Conference on Computer Aided Verification (CAV’22)*, Vol. 13372 of *Lecture Notes in Computer Science*, Springer, 2022, pp. 174–187. [path\(doi : 10.1007/978-3-031-13188-2\\_9\)](https://doi.org/10.1007/978-3-031-13188-2_9).
- [8] R. Bagnara, P. M. Hill, E. Zaffanella, 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. [path\(doi : 10.1016/j.scico.2007.08.001\)](https://doi.org/10.1016/j.scico.2007.08.001).
- [9] R. Koymans, Specifying real-time properties with metric temporal logic, *Real-time systems* 2 (4) (1990) 255–299.
- [10] R. Alur, T. Feder, T. A. Henzinger, The benefits of relaxing punctuality, *Journal of the ACM (JACM)* 43 (1) (1996) 116–146.
- [11] O. Maler, D. Nickovic, A. Pnueli, Checking temporal properties of discrete, timed and continuous behaviors, *Pillars of Computer Science: Essays Dedicated to Boris (Boaz) Trakhtenbrot on the Occasion of His 85th Birthday* (2008) 475–505.
- [12] R. Alur, D. Dill, A theory of timed automata, *Theoretical Computer Science* 126 (1994) 183–235.
- [13] R. Alur, A. Trivedi, D. Wojtczak, Optimal scheduling for constant-rate multi-mode systems, in: *Proceedings of the 15th ACM international conference on Hybrid Systems: Computation and Control*, 2012, pp. 75–84.
- [14] M. Blondin, P. Offtermatt, A. Sansfaçon-Buchanan, [Verifying linear temporal specifications of constant-rate multi-mode systems](https://doi.org/10.1109/LICS56636.2023.10175721), in: *LICS*, 2023, pp. 1–13. [path\(doi : 10.1109/LICS56636.2023.10175721\)](https://doi.org/10.1109/LICS56636.2023.10175721).  
URL <https://doi.org/10.1109/LICS56636.2023.10175721>

- [15] G. Frehse, C. L. Guernic, A. Donzé, S. Cotton, R. Ray, O. Lebeltel, R. Ripado, A. Girard, T. Dang, O. Maler, Spaceex: Scalable verification of hybrid systems, in: CAV 11: Proc. of 23rd Conf. on Computer Aided Verification, 2011, pp. 379–395.
- [16] M. Benerecetti, M. Faella, Automatic synthesis of switching controllers for linear hybrid systems: Reachability control, ACM Trans. on Embedded Computing Systems 16 (4) (2017).
- [17] M. Kloetzer, C. Belta, A fully automated framework for control of linear systems from temporal logic specifications, IEEE Transactions on Automatic Control 53 (1) (2008) 287–297.
- [18] K. Bae, J. Lee, [Bounded model checking of signal temporal logic properties using syntactic separation](#), Proc. ACM Program. Lang. 3 (POPL) (2019) 51:1–51:30. [path\(\*doi\* : 10.1145/3290364\)](#).  
URL <https://doi.org/10.1145/3290364>
- [19] J. Lee, G. Yu, K. Bae, [Efficient smt-based model checking for signal temporal logic](#), in: 36th IEEE/ACM International Conference on Automated Software Engineering, ASE 2021, Melbourne, Australia, November 15-19, 2021, IEEE, 2021, pp. 343–354. [path\(\*doi\* : 10.1109/ASE51524.2021.9678719\)](#).  
URL <https://doi.org/10.1109/ASE51524.2021.9678719>
- [20] R. Alur, T. Henzinger, P.-H. Ho, Automatic symbolic verification of embedded systems, IEEE Trans. Softw. Eng. 22 (1996) 181–201. [path\(\*doi\* : 10.1109/32.489079\)](#).
- [21] O. Maler, D. Nickovic, Monitoring temporal properties of continuous signals, in: Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems, Springer, 2004, pp. 152–166.
- [22] E. Davis, Infinite loops in finite time: Some observations, in: Proc. of the 3rd Int. Conf. on Principles of Knowledge Representation and Reasoning (KR'92). Cambridge, MA, USA, October 25-29, 1992, Morgan Kaufmann, 1992, pp. 47–58.
- [23] M. Benerecetti, M. Faella, S. Minopoli, Automatic synthesis of switching controllers for linear hybrid systems: Safety control, Theoretical Computer Science 493 (2013) 116–138.
- [24] D. Bremner, [Incremental convex hull algorithms are not output sensitive](#), Discret. Comput. Geom. 21 (1) (1999) 57–68. [path\(\*doi\* : 10.1007/PL00009410\)](#).  
URL <https://doi.org/10.1007/PL00009410>
- [25] A. Pnueli, The Temporal Logic of Programs, in: Foundation of Computer Science'77, IEEE Computer Society, 1977, pp. 46–57.
- [26] D. Dams, Flat Fragments of CTL and CTL\*: Separating the Expressive and Distinguishing Powers, Logic Journal of the IGPL 7 (1) (1999) 55–78.

- [27] G. D. Giacomo, M. Y. Vardi, [Linear temporal logic and linear dynamic logic on finite traces](#), in: F. Rossi (Ed.), IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013, IJCAI/AAAI, 2013, pp. 854–860.  
URL <http://www.aaai.org/ocs/index.php/IJCAI/IJCAI13/paper/view/6997>
- [28] A. Duret-Lutz, A. Lewkowicz, A. Fauchille, T. Michaud, E. Renault, L. Xu, Spot 2.0 – A framework for LTL and-automata manipulation, in: International Symposium on Automated Technology for Verification and Analysis, Springer, 2016, pp. 122–129.
- [29] V. Tramo, [Design and Implementation of a Model-Checking Tool for Verification of Linear Temporal Properties on Polyhedral Systems](#), Master’s thesis, Università degli Studi di Napoli Federico II, Napoli, Italy (2025).  
URL <https://github.com/vtramo/rtl-mc/blob/main/docs/thesis.pdf>
- [30] [RTL-MC - A Real-Time Logic Model Checker for Polyhedral Systems](#) (2025).  
URL <https://github.com/vtramo/rtl-mc>

## Appendix A. Additional Proofs

### Appendix A.1. Proofs from Section 2

**Lemma 2.** *For all polyhedra  $A$  and convex polyhedra  $B$ , the following holds:*

$$\text{reach}^0(A, B) = A \cap \text{cl}(B) \cap B_{\searrow_{>0}}.$$

*Proof.* Let  $\Delta \triangleq A \cap \text{cl}(B) \cap B_{\searrow_{>0}}$ . To show that  $\Delta \subseteq X^0$ , we just need to observe first that if  $x \in \Delta$ , then  $x \in A$ . Moreover,  $x \in B_{\searrow_{>0}}$ , which means that there is a direction  $d \in \text{Flow}$  such that  $x + d \cdot t \in B$ , for some  $t > 0$ . In addition, since  $x \in \text{cl}(B)$  and  $B$  is convex, we have that  $x + d \cdot t' \in B$ , for all  $t' \in (0, t]$ . But  $f(t) = x + d \cdot t$  is clearly an admissible trajectory that satisfies the required conditions for  $x$  to be in  $X^0$ . Hence, the conclusion.

For the other direction, let  $x \in X^0$ ,  $f$  be any admissible witness trajectory and  $t > 0$  be such that  $f(t') \in B$  for all  $t' \in (0, t]$ . Let, in addition,  $y \triangleq f(t)$ . By convexity of  $B$ , the segment connecting  $x$  and  $y$  lies entirely in  $\text{cl}(B)$ . Since  $y$  can be reached from  $x$  following an admissible trajectory, by convexity of  $\text{Flow}$  and Lemma 1, there is a direction  $d \in \text{Flow}$  such that  $y = x + d \cdot t$ . Clearly, the set

$$\{z \in \mathbb{R}^n \mid z = x + d \cdot t', \text{ for some } t' \in [0, t]\}$$

contains all and only the points of the segment from  $x$  to  $y$  and is, therefore, contained in  $\text{cl}(B)$ . Hence, we conclude that  $x \in A \cap \text{cl}(B) \cap B_{\searrow_{>0}}$ , as required. ■

**Lemma 3.** *For all polyhedra  $A$  and convex polyhedra  $B$ , the following holds:*

$$\text{reach}^+(A, B) = \bigcup_{P \in \text{Patch}(A)} \text{RWA}^m(T_P, \overline{A}), \quad \text{where } T_P \triangleq P \cap (\text{cl}(P) \cap B)_{\searrow_{>0}}.$$

*Proof.* First, observe that, by definition,  $T_P \subseteq P \subseteq A$ . As a consequence, we obtain that:

$$\begin{aligned} \text{RWA}^m(T_P, \overline{A}) &= \{x \in \mathbb{R}^n \mid \exists f \in \text{Traj}_{\text{wb}}(x), t \geq 0. \\ &\quad f(t) \in T_P \text{ and } \forall t' \in [0, t). f(t') \in A\}. \end{aligned}$$

Consider any point  $x \in \text{RWA}^m(T_P, \overline{A})$ , a trajectory  $f$  witnessing its membership to  $\text{RWA}^m(T_P, \overline{A})$ , a time instant  $t \in \mathbb{R}^+$  such that  $f(t) \in T_P$  and  $f(t') \in A$ , for all  $t' \in [0, t)$ , and let  $y \triangleq f(t) \in T_P$ . Clearly,  $y \in P$  and also  $y \in (\text{cl}(P) \cap B)_{\searrow_{>0}}$ . This means that there is an admissible straight trajectory  $f'$  that, in a strictly positive amount of time, leads from  $y$  to a point belonging both to the closure of  $P$  and to  $B$ . Let  $t^* > 0$  be a time instant such that  $f'(t^*) \in \text{cl}(P) \cap B$ . Since  $f'$  is a straight trajectory and  $P$  is a convex polyhedron,  $f'(t')$  is contained in  $P$ , hence also in  $A$ , for all  $t' \in [0, t^*)$ . By concatenating  $f$  with  $f'$  we obtain an admissible trajectory  $f''$  defined as follows:  $f''(t') = f(t')$ , for all  $t' \in [0, t]$ , and  $f''(t') = f'(t' - t)$ , for all  $t' \in (t, t + t^*]$ . Clearly,  $f''$  leads from  $x \in A$  to a point  $z \in B$ , while never leaving  $A$  except, possibly, in the last instant. In

addition,  $x = f(0) = f''(0)$  and  $f(0)$  is required to belong to  $A$ . By combining these observations, we obtain that for all  $x \in RWA^m(T_P, \bar{A})$  it holds that there exists  $f \in \text{Traj}_{\text{wb}}(x)$  and  $t \in \mathbb{R}_{>0}$  with  $f(t) \in B$  and for all  $t' \in (0, t)$ ,  $f(t') \in A$ . Hence, for all  $P \in \text{Patch}(A)$ , we have that  $RWA^m(T_P, \bar{A}) \subseteq \text{reach}^+(A, B)$ .

For the other direction, assume  $x \in \text{reach}^+(A, B)$ . Then there exist  $f \in \text{Traj}_{\text{wb}}(x)$  and  $t \in \mathbb{R}_{>0}$ , with  $f(t) \in B$  and  $f(t') \in A$ , for all  $t' \in (0, t)$ . Since  $A$  is a polyhedron and  $f$  is well-behaved, the trajectory can only change convex polyhedron in  $\text{Patch}(A)$  a finite number of times. This means that there is a last patch of  $A$  traversed by  $f$  before entering  $B$  in which  $f$  lingers for a positive amount of time. Let  $P$  be such a patch. Since as soon as  $f$  exits from  $P$  it enters  $B$ , it must do so by passing at time  $t$  through a point in  $B$  that lies on the border between  $P$  and  $B$ , that is  $f(t) \in \text{cl}(P) \cap B$ . Let  $t^* \in [0, t)$  be a time interval such that  $f(t') \in P$ , for all  $t' \in (t^*, t)$ . By convexity of  $P$  and  $\text{Flow}$  and thanks to Lemma 1, we obtain that  $f(t') \in P \cap (\text{cl}(P) \cap B)_{\angle < 0} = T_P$ , for all  $t' \in (t^*, t)$ . But then  $f$  is a witness of the membership of  $x$  to the set  $RWA^m(T_P, \bar{A})$ . As a consequence, we obtain that  $\text{reach}^+(A, B) \subseteq \bigcup_{P \in \text{Patch}(A)} RWA^m(T_P, \bar{A})$ . ■

#### Appendix A.2. Proofs from Section 4

**Lemma 4.** *Let  $\sigma : I \rightarrow 2^{AP}$  be a signal,  $\tau \in \text{TS}(\sigma)$  one of its time slicings,  $t \in I$  a time instant in the signal domain, and  $h = \text{slice}_{\sigma}^{\tau}(t)$  the corresponding slice index. Then, it holds true that:*

$$\text{trc}(\sigma_{>t}, \tau_{\geq t}) = \begin{cases} \text{trc}(\sigma, \tau)_{\geq h+1}, & \text{if } \text{sing} \in \text{trc}(\sigma, \tau)_h; \\ \text{trc}(\sigma, \tau)_{\geq h}, & \text{otherwise.} \end{cases}$$

*Proof.* Let  $\{t_i\}_{i=0} = \tau$ . In the following, we consider the case of a left-closed signal  $\sigma$  only, the left-open case being *mutatis mutandis* virtually the same.

- $[\text{sing} \in \text{trc}(\sigma, \tau)_h]$ . By definition of the *slice* and *trc* functions, there is a number  $l \in \mathbb{N}$  such that  $h = 2l$  and  $t = t_l$ . In addition,  $\tau_{\geq t} = \{t'_i\}_{i=0} = \{t_{i+l}\}_{i=0} \in \text{TS}(\sigma_{>t})$ . Now, let  $\{\alpha'_i\}_{i=0}$  be the sequence of observables such that, for all indices  $i$  and time instants  $t' \in (t'_i, t'_{i+1}) = (t_{i+l}, t_{i+l+1})$ , it holds true that  $\alpha'_i = \sigma_{>t}(t') = \sigma(t') = \alpha_{i+l}$ , where  $\{\alpha'_i\}_{i=0}$  is the sequence of observables associated with  $\tau$ . Hence, for all  $j \in \text{rng}(\text{slice}_{\sigma_{>t}}^{\tau_{\geq t}})$ , we have

$$\begin{aligned} \text{trc}(\sigma_{>t}, \tau_{\geq t})_j &\triangleq \begin{cases} \alpha'_i, & \text{if } j = 2i; \\ \sigma_{>t}(t'_i) \cup \{\text{sing}\}, & \text{if } j = 2i - 1; \end{cases} &= \\ &= \begin{cases} \alpha_{i+l}, & \text{if } j = 2i; \\ \sigma(t_{i+l}) \cup \{\text{sing}\}, & \text{if } j = 2i - 1; \end{cases} &= \\ &= \begin{cases} \alpha_i, & \text{if } j = 2i - 2l; \\ \sigma(t_i) \cup \{\text{sing}\}, & \text{if } j = 2i - 2l - 1; \end{cases} &= \\ &= \begin{cases} \alpha_i, & \text{if } j + h = 2i; \\ \sigma(t_i) \cup \{\text{sing}\}, & \text{if } j + h = 2i - 1; \end{cases} &= \end{aligned}$$

$$\begin{aligned}
&= \begin{cases} \alpha_i, & \text{if } j + h + 1 = 2i + 1; \\ \sigma(t_i) \cup \{\text{sing}\}, & \text{if } j + h + 1 = 2i; \end{cases} = \\
&= \text{trc}(\sigma, \tau)_{(j+h+1)} = (\text{trc}(\sigma, \tau)_{\geq h+1})_j,
\end{aligned}$$

which obviously implies  $\text{trc}(\sigma_{>t}, \tau_{\geq t}) = \text{trc}(\sigma, \tau)_{\geq h+1}$  as stated by the lemma.

- $[\text{sing} \notin \text{trc}(\sigma, \tau)_h]$ . By definition of the *slice* and *trc* functions, there is a number  $l \in \mathbb{N}$  such that  $h = 2l + 1$  and  $t \in (t_l, t_{l+1})$ . In addition,  $\tau_{\geq t} = \{t'_i\}_{i=0} = \{t\} \cup \{t_{i+l}\}_{i=1} \in TS(\sigma_{>t})$ . Now, let  $\{\alpha'_i\}_{i=0}$  be the sequence of observables such that  $\alpha'_0 = \sigma_{>t}(t') = \sigma(t') = \alpha_l$ , for all time instants  $t' \in (t'_0, t'_1) = (t, t_{l+1})$ , and  $\alpha'_i = \sigma_{>t}(t') = \sigma(t') = \alpha_{i+l}$ , for all indices  $i \geq 1$  and time instants  $t' \in (t'_i, t'_{i+1}) = (t_{i+l}, t_{i+l+1})$ . Hence, for all  $j \in \text{rng}(\text{slice}_{\sigma_{>t}}^{\tau_{\geq t}})$ , we have

$$\begin{aligned}
\text{trc}(\sigma_{>t}, \tau_{\geq t})_j &\triangleq \begin{cases} \alpha'_i, & \text{if } j = 2i; \\ \sigma_{>t}(t'_i) \cup \{\text{sing}\}, & \text{if } j = 2i - 1; \end{cases} = \\
&= \begin{cases} \alpha_{i+l}, & \text{if } j = 2i; \\ \sigma(t_{i+l}) \cup \{\text{sing}\}, & \text{if } j = 2i - 1; \end{cases} = \\
&= \begin{cases} \alpha_i, & \text{if } j = 2i - 2l; \\ \sigma(t_i) \cup \{\text{sing}\}, & \text{if } j = 2i - 2l - 1; \end{cases} = \\
&= \begin{cases} \alpha_i, & \text{if } j + h = 2i + 1; \\ \sigma(t_i) \cup \{\text{sing}\}, & \text{if } j + h = 2i; \end{cases} = \\
&= \text{trc}(\sigma, \tau)_{(j+h)} = (\text{trc}(\sigma, \tau)_{\geq h})_j,
\end{aligned}$$

which obviously implies  $\text{trc}(\sigma_{>t}, \tau_{\geq t}) = \text{trc}(\sigma, \tau)_{\geq h}$  as stated by the lemma. ■

**Lemma 5.** For all RTL formulae  $\varphi$ , signals  $\sigma: I \rightarrow 2^{AP}$ , and open intervals  $B \subseteq I$  such that  $\sigma$  is  $B$ -uniform, the following holds true:  $\sigma_{\sim_1 t_1} \models \varphi$  iff  $\sigma_{\sim_2 t_2} \models \varphi$ , for all  $t_1, t_2 \in B$  and  $\sim_1, \sim_2 \in \{\geq, >\}$ .

*Proof.* The proof proceeds by structural induction on the RTL formula  $\varphi$ . Since the Boolean operators  $\neg$ ,  $\wedge$ , and  $\vee$  are trivial to deal with and due to the duality property between  $\mathbf{U}$  and  $\mathbf{R}$ , we can focus on the atomic propositions and the two temporal operators  $\mathbf{X}$  and  $\mathbf{U}$  only. In the following, *w.l.o.g.*, let us assume  $\sigma_{\sim_1 t_1} \models \varphi$ .

- **[Base case  $\varphi = p \in AP$ ].** It is easy to see that there necessarily exists  $t \in B$  such that  $p \in \sigma(t)$ . Indeed, if  $\sim_1 = \geq$ , by the semantics of atomic propositions on left-closed signals, we can choose  $t = t_1$ , since  $p \in \sigma_{\geq t_1}(t_1) = \sigma(t_1)$ . If  $\sim_1 = >$ , instead, again by the semantics of atomic propositions, this time for left-open signals, there exists a non-empty open interval  $(t_1, t') \subseteq B$  such that  $p \in \sigma_{>t_1}(t'') = \sigma(t'')$ , for all  $t'' \in (t_1, t')$ . Since, by hypothesis,  $B$  is a non-empty open interval with  $t_1 \in B$ , the intersection  $B \cap (t_1, t')$  is non-empty as well. Therefore, we can arbitrarily choose  $t$  as an element of this intersection.

At this point, consider the left-closed subinterval  $C \triangleq [t_2, \sup(B))$  of  $B$ . Due to the  $B$ -uniformity of the signal  $\sigma$ , it holds that  $p \in \sigma(t') = \sigma(t)$ , for all  $t' \in C$ . Hence, by using  $C$  as witness, it is immediate to show that  $\sigma_{\sim_2 t_2} \models \varphi$ , independently from the specific relation  $\sim_2$ .

- **[Inductive case  $\varphi = X\varphi'$ ].** Independently from the relation  $\sim_1$ , by definition of the semantics of the temporal operator  $X$ , there exists  $t \in (t_1, \sup(I)]$  such that  $\sigma_{\geq t'} \models \varphi$ , for all  $t' \in (t_1, t)$ . Since, by hypothesis,  $B$  is an open interval and  $t_1 \in B$ , the intersection  $B \cap (t_1, t)$  is necessarily non-empty. Thus, there exists an instant  $t' \in B$  such that  $\sigma_{\geq t'} \models \varphi$ . Therefore, by the inductive hypothesis, it holds that  $\sigma_{\geq t'} \models \varphi$ , for all  $t' \in B$ . Now, since  $t_2 \in B$ , also the intersection  $B \cap (t_2, \sup(I))$  is non-empty. Hence, there exists  $t \in (t_2, \sup(I)]$  such that  $\sigma_{\geq t'} \models \varphi$ , for all  $t' \in (t_2, t)$ , which implies, again due to the semantics of the temporal operator, that  $\sigma_{\sim_2 t_2} \models \varphi$ .
- **[Inductive case  $\varphi = \varphi_1 \cup \varphi_2$ ].** Independently from the relation  $\sim_1$ , by definition of the semantics of the temporal operator  $\cup$ , there exists  $t \in \text{dom}(\sigma_{\sim_1 t_1})$  such that  $\sigma_{\geq t} \models \varphi_2$  and  $\sigma_{\geq t'} \models \varphi_1$ , for all  $t' \in \text{dom}(\sigma_{\sim_1 t_1})$  with  $t' < t$ . Since, by hypothesis,  $B$  is an open interval and  $t_1 \in B$ , the intersection  $B \cap \text{dom}(\sigma_{\sim_1 t_1})$  is necessarily non-empty. Thus, there exists an instant  $t' \in B$  such that  $\sigma_{\geq t'} \models \varphi_1$ . Hence, by the inductive hypothesis applied to the formula  $\varphi_1$ , it holds that  $\sigma_{\geq t'} \models \varphi_1$ , for all  $t' \in B$ . Now, two cases may arise depending on whether  $t$  belongs to  $B$  as well.
  - **[ $t < \sup(B)$ ].** Since  $\sigma_{\geq t} \models \varphi_2$  and  $t \in B$ , by the inductive hypothesis applied to the formula  $\varphi_2$ , it holds that  $\sigma_{\geq t'} \models \varphi_2$ , for all  $t' \in B$ . Then, as an immediate consequence, any  $t \in (t_2, \sup(B))$  satisfies both  $\sigma_{\geq t} \models \varphi_2$  and  $\sigma_{\geq t'} \models \varphi_1$ , for all  $t' \in (t_2, t)$ . Hence,  $\sigma_{\sim_2 t_2} \models \varphi$  holds, independently from the specific relation  $\sim_2$ .
  - **[ $t \geq \sup(B)$ ].** Since  $t_2 \in B$ , it holds that  $t_2 < t$ . Hence, to prove that  $\sigma_{\sim_2 t_2} \models \varphi$ , independently from the specific relation  $\sim_2$ , it only remains to show that  $\sigma_{\geq t'} \models \varphi_1$ , for all  $t' \in [t_2, t)$ . Obviously, the interval  $[t_2, t)$  can be decomposed into the disjoint union of the two adjacent intervals  $[t_2, \sup(B))$  and  $[\sup(B), t)$ . At this point, the required property clearly follows from the fact that  $[t_2, \sup(B)) \subset B$  and  $[\sup(B), t) \subset (t_1, t)$ , as  $\varphi_1$  holds true on all points of both intervals. ■

**Lemma 6.** *For all RTL formulae  $\varphi$ , signals  $\sigma: I \rightarrow 2^{AP}$ , and time instants  $t \in I$ , the following holds true:  $\sigma_{>t} \models \varphi$  iff there exists a time instant  $t' \in I$ , with  $t' > t$ , such that  $\sigma_{\geq t''} \models \varphi$ , for all  $t'' \in (t, t']$ .*

*Proof.* The proof proceeds by induction on the Boolean structure of the  $\text{RTL}_f$  formula  $\varphi$ , where we consider as base cases the atomic propositions and the two temporal operators  $X$  and  $\cup$ . Since the inductive cases of Boolean operators  $\neg$ ,  $\wedge$ , and  $\vee$  are trivial to deal with and thanks to the duality property between  $\cup$  and  $\text{R}$ , here we focus on the base cases only.

- **[Base case  $\varphi = p \in AP$ ].** Since  $\sigma_{>t}$  is a left-open signal, by the semantic of atomic propositions, we have that  $\sigma_{>t} \models p$  holds *iff* there exists a non-empty interval  $(t, t'] \subseteq I$  such that  $p \in \sigma_{>t}(t'') = \sigma(t'')$ , for all  $t'' \in (t, t']$ , which also means  $\sigma_{\geq t''} \models p$ , again by the semantic of atomic propositions, this time on left-closed signals. Hence, the truth of the statement is immediately verified.
- **[Base case  $\varphi = X\varphi'$ ].** By the semantics of the temporal operator  $X$ , we have that  $\sigma_{>t} \models X\varphi'$  holds *iff* there exists a non-empty interval  $(t, t'] \subseteq I$  such that  $\sigma_{\geq t''} \models \varphi'$ , for all  $t'' \in (t, t']$ . As an immediate consequence, by using  $t'$  as witness of the second property, again by the semantic of the temporal operator, we have that the previous statement is equivalent to say that  $\sigma_{\geq t''} \models X\varphi'$ , for all  $t'' \in (t, t']$ .
- **[Base case  $\varphi = \varphi_1 \cup \varphi_2$ , only-if direction].** By the semantics of the temporal operator  $\cup$ , if  $\sigma_{>t} \models \varphi_1 \cup \varphi_2$ , then there exists  $t_2 \in I$  such that  $\sigma_{\geq t_2} \models \varphi_2$  and  $\sigma_{\geq t_1} \models \varphi_1$ , for all  $t_1 \in (t, t_2)$ . As an immediate consequence, by using precisely  $t' \triangleq t_2$  as witness of the second property, we have that  $\sigma_{\geq t''} \models \varphi_1 \cup \varphi_2$ , for all  $t'' \in (t, t']$ .
- **[Base case  $\varphi = \varphi_1 \cup \varphi_2$ , if direction].** Let  $\tau = \{t_i\}_{i=0} \in TS(\sigma_{>t})$  be a time slicing of the suffix  $\sigma_{>t}$  of the signal  $\sigma$  and  $j \geq 1$  the smallest index such that either (a)  $\sigma_{\geq t_j} \models \varphi_2$  or (b)  $\sigma_{\geq t''} \models \varphi_2$ , for all  $t'' \in (t_{j-1}, t_j)$ . The existence of such an index is ensured by the fact that at every time instant  $t''$  of the non-empty interval  $(t, t']$  the until formula  $\varphi_1 \cup \varphi_2$  is satisfied. Now, suppose by contradiction that  $\sigma_{>t} \not\models \varphi_1 \cup \varphi_2$ . Since  $\varphi_2$  is satisfied either at time instant  $t_j$  or at all time instants  $t''$ , with  $t'' \in (t_{j-1}, t_j)$ , the only possibility for the until formula to be falsified is the existence of at least one time instant  $t'$ , either in  $(t, t_j)$  or in  $(t, t_{j-1}]$ , such that  $\sigma_{\geq t'} \not\models \varphi_1$ . However, this would clearly lead to  $\sigma_{\geq t''} \not\models \varphi_1 \cup \varphi_2$ , for all  $t'' \in (t, t']$ , which contradicts the hypothesis. ■

**Lemma 7.** For all RTL formulae  $\varphi$  and signals  $\sigma: I \rightarrow 2^{AP}$ , the following holds true:  $\sigma \models X\varphi$  iff  $\sigma_{>\inf(I)} \models \varphi$ .

*Proof.* By the semantics of the temporal operator  $X$ , it holds that  $\sigma \models X\varphi$  *iff* there exists  $t \in I$  with  $t > \inf(I)$  such that  $\sigma_{\geq t'} \models \varphi'$ , for all  $t' \in (\inf(I), t]$ . By Lemma 6, the latter is precisely equivalent to  $\sigma_{>\inf(I)} \models \varphi$ . ■

### Appendix A.3. Proofs from Section 5

**Lemma 8.** Assume that  $\mathbf{0} \in \text{int}(\text{Flow})$ . Let  $P$  be a tile and  $x$  and  $y$  be any two points in  $P$ . Then there exists an admissible trajectory  $f$  and a time delay  $\delta \geq 0$  such that  $f(0) = x$ ,  $f(\delta) = y$ , and  $f(t) \in P$ , for all  $t \in (0, \delta)$ .

*Proof.* Assume  $x \in A$  and  $y \in B$ , where  $A$  and  $B$  are patches of tile  $P$ . Since the patches in  $P$  are connected, there is a sequence  $C_0, C_1, \dots, C_n$  of patches of  $P$ , such that  $C_0 = A$ ,  $C_n = B$  and  $C_i$  is adjacent to  $C_{i+1}$ , for all  $i \in \{0, \dots, n-1\}$ . To obtain a trajectory from  $x$  to  $y$ , we first select points  $x_1, \dots, x_n$  such that each  $x_i$  belongs to the border between  $C_{i-1}$  and  $C_i$ . Then let  $f_0$  be the straight

segment connecting  $x$  to  $x_1$ ,  $f_n$  be the straight segment connecting  $x_n$  to  $y$  and, for each  $i \in \{1, \dots, n-1\}$ , let  $f_i$  be the straight segment connecting  $x_i$  to  $x_{i+1}$ . Since  $\mathbf{0} \in \text{int}(Flow)$ , for each straight segment  $f_i$ , there exists a vector  $v_i \in Flow$  and a duration  $t_i > 0$ , such that  $x_{i+1} = x_i + v_i \cdot t_i$  and  $f_i(t) = x_i + v_i \cdot t$ , for all  $t \in [0, t_i]$ . Observe that each such straight segment is contained in the tile  $P$  the patches belong to. Concatenating those straight segment together gives us the desired trajectory from  $x$  to  $y$ . To this end, we define a trajectory  $f$  as follows. First let  $\delta_i = \sum_{j=0}^{i-1} t_j$ , for each  $0 \leq i \leq n$ . Then, we set  $f(t) = f_i(t - \delta_i)$ , if  $t \in [\delta_i, \delta_{i+1})$ , for some  $0 \leq i \leq n$ . The trajectory  $f$  so defined is admissible and connects  $x$  with  $y$  without ever exiting  $P$ . In addition, it requires time  $\delta \triangleq \delta_n$  to go from  $x$  to  $y$ , *i.e.*,  $f(0) = x$  and  $f(\delta) = y$ . ■

#### Appendix A.4. Proofs from Section 6

Recall that, thanks to the equivalence  $\varphi_1 \mathbf{R} \varphi_2 \equiv (\mathbf{G} \varphi_2) \vee (\varphi_2 \mathbf{U} (\varphi_1 \wedge \varphi_2))$ , we can always transform an RTL formula into an equivalent one, where no release temporal operators occur. Moreover, in a non-recurrent formula, all arguments of a temporal operator  $\mathbf{G}$  are propositional. We can therefore prove the following two claims by restricting to such non-recurrent RTL formulae.

**Claim 1.** *Let  $\varphi$  be a non-recurrent RTL formula and  $\sigma: I \rightarrow 2^{AP}$  an infinite-time signal such that  $\sigma \models \varphi$ . Then, there exist a propositional formula  $\tilde{\eta}$  and a time instant  $\tilde{t} \in I$  with the following properties:*

- 1)  $\sigma_{\geq \tilde{t}} \models \mathbf{G} \tilde{\eta}$ ;
- 2) for all signals  $\sigma'$  with  $\sigma'_{\leq \tilde{t}} = \sigma_{\leq \tilde{t}}$  and  $\sigma'_{\geq \tilde{t}} \models \mathbf{G} \tilde{\eta}$ , it holds that  $\sigma' \models \varphi$ .

*Proof.* We shall prove the following stronger version of the claim, from which the original one immediately follows once  $t = t' = \inf(I)$  and  $\bar{t} = \tilde{t}$  are assumed.

Let  $\varphi$  be a non-recurrent RTL formula,  $\sigma: I \rightarrow 2^{AP}$  an infinite-time signal, and  $t \in \text{cl}(I)$  a time instant. Then, there exist a propositional formula  $\tilde{\eta}$  and a time instant  $\tilde{t} \in I$  with the following properties:

- 1) if  $\sigma_{\geq t'} \models \varphi$ , for some  $t' \in \text{cl}(I)$  with  $t' \leq t$ , then  $\sigma_{\geq \tilde{t}} \models \mathbf{G} \tilde{\eta}$ ;
- 2) for all time instants  $t' \in \text{cl}(I)$ , with  $t' \leq t$  and  $\sigma_{\geq t'} \models \varphi$ , time instants  $\bar{t} \in I$ , with  $\bar{t} \geq \tilde{t}$ , and signals  $\sigma'$  with  $\sigma'_{\leq \bar{t}} = \sigma_{\leq \bar{t}}$  and  $\sigma'_{\geq \bar{t}} \models \mathbf{G} \tilde{\eta}$ , it holds that  $\sigma'_{\geq t'} \models \varphi$ .

The proof proceeds by induction on the structure of the formula in positive normal form, where we consider the literals  $p$  and  $\neg p$  and the formulae  $\mathbf{G} \eta$ , with  $\eta$  a propositional formula, as base cases.

Before starting with the case analysis on  $\varphi$ , let us fix a signal  $\sigma: I \rightarrow 2^{AP}$  and an arbitrary time instant  $t \in \text{cl}(I)$ .

- **[Base case  $\varphi = p$  or  $\varphi = \neg p$ ]:** We assign  $\tilde{\eta} \triangleq \top$  and  $\tilde{t} \triangleq t + \epsilon$ , for some arbitrary value  $\epsilon > 0$ . One can immediately verify that the thesis holds true.

- **[Base case  $\varphi = \mathbf{G}\eta$ ]:** We assign  $\tilde{\eta} \triangleq \eta$  and  $\tilde{t} \triangleq t + \epsilon$ , for some arbitrary value  $\epsilon > 0$ . One can immediately verify that the thesis holds true.

- **[Inductive case  $\varphi = \varphi_1 \wedge \varphi_2$ ]:** By the inductive hypothesis applied to the two formulae  $\varphi_1$  and  $\varphi_2$  and the time instant  $t$ , there are two propositional formulae  $\tilde{\eta}_1$  and  $\tilde{\eta}_2$  and two time instants  $\tilde{t}_1, \tilde{t}_2 \in I$  such that

- + if  $\sigma_{\geq t'} \models \varphi_i$ , for some  $t' \in cl(I)$  with  $t' \leq t$ , then  $\sigma_{\geq \tilde{t}_i} \models \mathbf{G}\tilde{\eta}_i$  and
- + for all time instants  $t' \in cl(I)$ , with  $t' \leq t$  and  $\sigma_{\geq t'} \models \varphi_i$ , time instants  $\bar{t} \in I$ , with  $\bar{t} \geq \tilde{t}_i$ , and signals  $\sigma'$  with  $\sigma'_{\leq \bar{t}} = \sigma_{\leq \bar{t}}$  and  $\sigma'_{\geq \bar{t}} \models \mathbf{G}\tilde{\eta}_i$ , it holds that  $\sigma'_{\geq t'} \models \varphi_i$ ,

for both  $i \in \{1, 2\}$ . We now choose  $\tilde{\eta} \triangleq \tilde{\eta}_1 \wedge \tilde{\eta}_2$  and  $\tilde{t} \triangleq \max\{\tilde{t}_1, \tilde{t}_2\}$ . At this point, one can easily verify that the thesis holds true.

- **[Inductive case  $\varphi = \varphi_1 \vee \varphi_2$ ]:** By the inductive hypothesis applied to the two formulae  $\varphi_1$  and  $\varphi_2$  and the time instant  $t$ , there are two propositional formulae  $\tilde{\eta}_1$  and  $\tilde{\eta}_2$  and two time instants  $\tilde{t}_1, \tilde{t}_2 \in I$  such that

- + if  $\sigma_{\geq t'} \models \varphi_i$ , for some  $t' \in cl(I)$  with  $t' \leq t$ , then  $\sigma_{\geq \tilde{t}_i} \models \mathbf{G}\tilde{\eta}_i$  and
- + for all time instants  $t' \in cl(I)$ , with  $t' \leq t$  and  $\sigma_{\geq t'} \models \varphi_i$ , time instants  $\bar{t} \in I$ , with  $\bar{t} \geq \tilde{t}_i$ , and signals  $\sigma'$  with  $\sigma'_{\leq \bar{t}} = \sigma_{\leq \bar{t}}$  and  $\sigma'_{\geq \bar{t}} \models \mathbf{G}\tilde{\eta}_i$ , it holds that  $\sigma'_{\geq t'} \models \varphi_i$ ,

for both  $i \in \{1, 2\}$ . Now, if there is  $i \in \{1, 2\}$  such that (i)  $\sigma_{\geq t'} \models \varphi_i$ , for some  $t' \in cl(I)$  with  $t' \leq t$ , but (ii)  $\sigma_{\geq t'} \not\models \varphi_{3-i}$ , for all  $t' \in cl(I)$  with  $t' \leq t$ , we choose  $\tilde{\eta} \triangleq \tilde{\eta}_i$  and  $\tilde{t} \triangleq \tilde{t}_i$ . Otherwise, we choose  $\tilde{\eta} \triangleq \tilde{\eta}_1 \wedge \tilde{\eta}_2$  and  $\tilde{t} \triangleq \max\{\tilde{t}_1, \tilde{t}_2\}$ . At this point, one can easily verify that the thesis holds true.

- **[Inductive case  $\varphi = \mathbf{X}\varphi'$ ]:** Consider a time instant  $t^* \triangleq t + \epsilon$ , for some arbitrary value  $\epsilon > 0$ . By the inductive hypothesis applied to the formula  $\varphi'$  and the time instant  $t^*$ , there exist a propositional formula  $\tilde{\eta}$  a time instant  $\tilde{t} \in I$  such that

- + if  $\sigma_{\geq t'} \models \varphi'$ , for some  $t' \in cl(I)$  with  $t' \leq t^*$ , then  $\sigma_{\geq \tilde{t}} \models \mathbf{G}\tilde{\eta}$  and
- + for all time instants  $t' \in cl(I)$ , with  $t' \leq t^*$  and  $\sigma_{\geq t'} \models \varphi'$ , time instants  $\bar{t} \in I$ , with  $\bar{t} \geq \tilde{t}$ , and signals  $\sigma'$  with  $\sigma'_{\leq \bar{t}} = \sigma_{\leq \bar{t}}$  and  $\sigma'_{\geq \bar{t}} \models \mathbf{G}\tilde{\eta}$ , it holds that  $\sigma'_{\geq t'} \models \varphi'$ .

At this point, using the same propositional formula  $\tilde{\eta}$  and time instant  $\tilde{t}$ , one can easily verify that the thesis holds true.

- **[Inductive case  $\varphi = \varphi_1 \mathbf{U} \varphi_2$ ]:** Let  $B \triangleq \{t' \leq t \mid \sigma_{\geq t'} \models \varphi\}$  and  $B_i \triangleq \{t' \leq t \mid \sigma_{\geq t'} \models \varphi_i\}$ . By definition of the semantics of the temporal operator  $\mathbf{U}$ , we clearly have  $B_2 \subseteq B$  and  $B \setminus B_2 \subseteq B_1$ . If  $\sup(B_2) = \sup(B)$ , let  $t^* \triangleq t$ . Otherwise,  $\sup(B_2) < \sup(B)$ . Consider an arbitrary instant  $t' \in B \setminus B_2$  with  $t' > \sup(B_2)$ . By the semantics of the  $\mathbf{U}$  operator, there exists a time instant  $t_2 > t$  such that  $\sigma_{\geq t_2} \models \varphi_2$  and, for all time instants  $t' \leq t_1 < t_2$ , it

holds that  $\sigma_{\geq t_1} \models \varphi_1$ . In this case, let  $t^* \triangleq t_2$ . By the inductive hypothesis applied to the two formulae  $\varphi_1$  and  $\varphi_2$  and the time instant  $t^*$ , there are two propositional formulae  $\tilde{\eta}_1$  and  $\tilde{\eta}_2$  and two time instants  $\tilde{t}_1, \tilde{t}_2 \in I$  such that

- + if  $\sigma_{\geq t'} \models \varphi_i$ , for some  $t' \in cl(I)$  with  $t' \leq t$ , then  $\sigma_{\geq \tilde{t}_i} \models \mathbf{G}\tilde{\eta}_i$  and
- + for all time instants  $t' \in cl(I)$ , with  $t' \leq t$  and  $\sigma_{\geq t'} \models \varphi_i$ , time instants  $\tilde{t} \in I$ , with  $\tilde{t} \geq \tilde{t}_i$ , and signals  $\sigma'$  with  $\sigma'_{\leq \tilde{t}} = \sigma_{\leq \tilde{t}}$  and  $\sigma'_{\geq \tilde{t}} \models \mathbf{G}\tilde{\eta}_i$ , it holds that  $\sigma'_{\geq t'} \models \varphi_i$ ,

for both  $i \in \{1, 2\}$ . Now, we choose  $\tilde{\eta} \triangleq \tilde{\eta}_2$  and  $\tilde{t} \triangleq \tilde{t}_2$ , if  $B = B_2$ , and  $\tilde{\eta} \triangleq \tilde{\eta}_1 \wedge \tilde{\eta}_2$  and  $\tilde{t} \triangleq \max\{\tilde{t}_1, \tilde{t}_2\}$ , otherwise. At this point, one can easily verify that the thesis holds true. ■

**Claim 2.** *Let  $\varphi$  be a non-recurrent RTL formula and  $\sigma: I \rightarrow 2^{AP}$  a finite-time signal such that  $\sigma \models \varphi$ . Then, for the unique infinite-time signal  $\sigma'$  with  $\sigma'(t) = \sigma(t)$ , for  $t \in I$ , and  $\sigma'(t) = \sigma(\sup(I))$ , otherwise, it holds that  $\sigma' \models \varphi$ .*

*Proof.* The proof proceeds by induction on the structure of the formula in positive normal form, where we consider the literals  $p$  and  $\neg p$  and the formulae  $\mathbf{G}\eta$ , with  $\eta$  a propositional formula, as base cases.

- **[Base case  $\varphi = p$  or  $\varphi = \neg p$ ]:** Clearly, the truth value of  $\varphi$  only depends on the initial portion of the signal, which is left unchanged by the operation of infinite extension, thus, if  $\sigma \models \psi$ , we immediately have  $\sigma' \models \psi$ .
- **[Base case  $\varphi = \mathbf{G}\eta$ ]:** By the semantic of the temporal operator  $\mathbf{G}$ , we have that, if  $\sigma \models \psi$ , we surely have  $\sigma_{\geq \sup(I)} \models \eta$ , which means that the set of atomic propositions  $\sigma(\sup(I))$  satisfies the propositional formula  $\eta$ . At this point, it is clear that, by definition,  $\sigma'_{\geq \sup(I)} \models \mathbf{G}\eta$ , from which it immediately follows that  $\sigma' \models \psi$ .
- **[Inductive cases]:** The thesis immediately follows by analysing the semantics of the operators and applying the inductive hypothesis to the subformulae. ■

#### Appendix A.5. Proofs from Section 8

**Lemma 13.** *For all hybrid runs  $\rho$ , states  $s \in S_{open}$ , and patches  $P \in Patch(\llbracket s \rrbracket)$ , there exists a hybrid run  $\rho'$  such that:*

- $\rho'$  starts and ends in the same pairs as  $\rho$ ;
- $\rho'$  passes at most once through the pair  $(P, s)$ ;
- $Visited(\rho') \subseteq Visited(\rho)$ ;
- the length of the shortest discrete trace of  $\rho'$  is smaller than or equal to that of  $\rho$ .

*Proof.* Assume that  $\rho$  passes at least twice through the pair  $(P, s)$ . Let  $t_1, t_2$  be two times in the domain of  $\rho$  belonging to the first and to the last visit to  $(P, s)$ . In particular,  $\rho(t_i) \in (P, s)$ , for  $i = 1, 2$ . Let  $\rho(t_i) = (x_i, s)$ , we define a new hybrid run  $\rho'$ , by connecting with a straight trajectory point  $x_1$  to  $x_2$ . By convexity of the flow, such a trajectory is feasible and, by convexity of  $P$ , it is also entirely contained in  $P$ . The rest of  $\rho'$  follows exactly  $\rho$ . It is easy to see that  $\rho'$  satisfies all properties required by the thesis. ■

**Lemma 14.** *For all states  $s \in S$ , convex polyhedra  $P \in \text{Patch}(\llbracket s \rrbracket)$  and  $X \subseteq P$ , and maps  $V : S \rightarrow_s 2^{\text{Patch}(\llbracket s \rrbracket)}$  such that  $P \notin V(s)$ , we have that  $\exists \text{Denot}(s, P, X, V)$  returns the set of all points  $x$  from which there is a hybrid run  $\rho \in \text{HRun}(x)$  such that: (a)  $\rho$  ends in  $(X, s)$ ; (b)  $\rho$  avoids  $V$ ; (c) if  $s \in S_{\text{open}}$ , then  $\rho$  avoids  $(P, s)$ , except for the last slice.*

*Proof.* [**Soundness**] First, we prove that the base case of the algorithm is sound, that is, that the points returned at Line 1 satisfy the lemma items. Any accepting initial state of  $\mathcal{A}_\varphi^{\text{fin}}$  is in itself a run of  $\mathcal{A}_\varphi^{\text{fin}}$  of length 1 from an initial state. If  $s \in S_{\text{sing}}$ , then the algorithm returns  $X = \text{reach}^0(X, \mathbb{R}^n)$ . For all  $x \in X$ , let  $f$  be the trajectory of duration 0 defined by  $f(0) = x$ . Its discrete trace  $w_f$  contains a single symbol and induces the run  $r^d = s$  in  $\mathcal{A}_\varphi^{\text{fin}}$ . The hybrid run  $(f, r^c)$  ends in  $(X, s)$ , giving Item (a); and avoids  $V$ , because its only point is  $(x, s)$  and, by assumption,  $x \in P \notin V(s)$ . Thus, we have Item b. Item (c) trivially holds since  $s \notin S_{\text{open}}$ .

If, on the other hand,  $s \notin S_{\text{open}}$ , then the algorithm returns  $\hat{X} \triangleq \text{reach}^+(X, \mathbb{R}^n)$ . Hence, for each point  $x \in \hat{X}$ , there exists a trajectory  $f$  and a time  $t > 0$  such that  $f(0) = x$  and  $f(t) \in X$ , for  $t' \in (0, t]$ . Its discrete trace  $w_f$  contains a single symbol and induces the run  $r^d = s$  in  $\mathcal{A}_\varphi^{\text{fin}}$ . The hybrid run  $(f, r^c)$  ends in  $(X, s)$ , giving Item (a); and avoids  $V$ , because its only points are in  $P$  and, by assumption,  $x \in P \notin V(s)$ . Thus, we have Item b. Moreover, each point in  $\hat{X}$  can remain for a while in  $X$  and, therefore, Item (c) holds as well.

Let us now consider the points added to the result at Line 9. We proceed by induction on the length  $k$  of the longest sequence of pairs  $(s_i, P_i)_{i=0, \dots, k-1}$  such that: (i) the sequence  $(s_i)_{i=0, \dots, k-1}$  is a (not necessarily initial) run of  $\mathcal{A}_\varphi^{\text{fin}}$  ending in  $s_{k-1} = s$ , (ii)  $P_{k-1} = P$ , (iii) each  $P_i$  is a patch of  $\llbracket s_i \rrbracket$ , (iv) if  $s_i \in S_{\text{open}}$  then  $P_i \notin V(s_i)$ , (v) all the pairs  $(s_i, P_i)$  such that  $s_i \in S_{\text{open}}$  are distinct. Note that Items (i) and (v) imply that the length of these sequences is bounded by twice the number of distinct non-singular pairs. We call the length so defined  $k(s, P, V)$ .

In the base case,  $k(s, P, V) = 1$ . Then, the algorithm does not perform any recursive call, because for each predecessor  $s'$  of  $s$ , the set  $A' \triangleq \text{reach}^{\text{type}(s')}(A, X)$  computed at Line 6 is empty, with  $A \triangleq \llbracket s' \rrbracket \setminus V(s')$ . Indeed, assume by contradiction, that there is a predecessor  $s'$  of  $s$  whose set  $A'$  is not empty. Therefore, the must be a pair  $(Q, Y) \in \text{split}(A', A)$ , where  $Q$  is a patch of  $A$  and the sequence  $(s', Q)(s, P)$  has all the properties (i)–(v) needed to prove that  $k(s, P, V) > 1$ , contradicting the assumption of the base case. We conclude that either  $s$  has no predecessors, or  $A'$  is empty. Hence, no points are added to the result at Line 9.

For the inductive case, assume that the longest sequence described above has length greater than 1. Let  $s'$  be a predecessor of  $s$  and let  $\{(Q_1, Y_1), \dots, (Q_n, Y_n)\}$  be  $\text{split}(A', A)$ . For all  $i = 1, \dots, n$ , we apply the inductive hypothesis to  $s'$ ,  $Q_i$ , and  $V'$ , where  $V' = V[s \mapsto V(s) \cup \{P\}]$ , if  $s \in S_{\text{open}}$ , and  $V' = V$ , otherwise, as prescribed by Line 3. In order to apply the inductive hypothesis, we prove that  $k(s', Q_i, V') < k(s, P, V)$  in both cases. Assume by contradiction that  $h \triangleq k(s', Q_i, V') \geq k(s, P, V)$  and let  $\xi \triangleq (s_i, P_i)_{i=0, \dots, h-1}$  be the sequence of length  $h$  corresponding to  $(s', Q_i, V')$ . We extend  $\xi'$  with the pair  $(s, P)$ , thus obtaining the sequence  $\xi' \triangleq \xi \cdot (s, P)$  of length  $h + 1$ . We can show that  $\xi'$  satisfies all five Items (i)-(v) w.r.t.  $(s, P, V)$ . Items (i)-(iii) are trivially true, so we can focus on the remaining two:

- If  $s \in S_{\text{open}}$ , then Item (iv) follows from the assumption that  $P \notin V(s)$ , while Item (v) is due to the fact that no pair in  $\xi$  can be equal to  $(s, P)$ , since  $P \in V'(s)$ .
- If  $s \in S_{\text{sing}}$ , then both Items (iv) and (v) hold trivially.

Hence,  $\xi'$  is a sequence satisfying (i)-(v) w.r.t.  $(s, P, V)$ , so  $k(s, P, V) \geq h + 1$ , which contradicts the hypothesis.

Now, consider a point  $x$  in  $\exists \text{Denot}(s', Q_i, Y_i, V')$  and the witness hybrid run  $\rho' = (f', r')$  provided by the inductive hypothesis, whose trajectory  $f'$  goes from  $x$  to  $Y_i \subseteq A$ , and let  $\{t_j\}_{j=0}^k$  be its time slicing. In the following we shall extend  $\rho'$  to reach  $(X, s)$ , using the definition of  $\text{reach}$ , while satisfying the Items (a), (b), and (c) of the lemma. We again distinguish two cases.

- [ $s' \in S_{\text{sing}}$ ] By Proposition 1(a),  $\text{type}(s') = 0$  and  $s \in S_{\text{open}}$ . In this case,  $f'$  must end in some point  $z \in Y_i$ . Since  $Y_i \subseteq A' = \text{reach}^0(A, X)$ , let  $f''$  be the trajectory that starts in  $z$ , immediately enters  $X$ , and remains inside  $X$  in the interval  $(0, \epsilon)$ , for some  $\epsilon > 0$ . Let  $f$  be the concatenation of  $f'$  and  $f''$ . It is immediate to observe that  $\tau = \{t_j\}_{j=0}^{k+1}$ , with  $t_{k+1} = (t_k + \epsilon)$ , is a time slicing of  $f$ .

Let us set  $\rho \triangleq (f, r)$ , where  $r$  is the continuous run of  $f$  that has the following form:

$$r(t) = \begin{cases} r'(t) & \text{if } 0 \leq t \leq t_k \\ s & \text{if } t_k < t < t_k + \epsilon. \end{cases}$$

Clearly,  $\rho$  is a hybrid run in  $\text{HRun}(x)$  with  $\tau$  one of its time slicings.

As, by construction,  $\rho$  ends in  $(X, s)$ , we obtain that Item (a) holds. Item (b) is satisfied, since  $\rho'$  avoids  $V'$ , by assumption  $P \notin V(s)$ , and  $V$  is pointwise included in  $V'$ . Item (c), instead, follows from the fact that  $V' = V[s \mapsto V(s) \cup P]$ .

- [ $s' \in S_{\text{open}}$ ] Obviously,  $\text{type}(s') = +$ , and  $s \in S_{\text{sing}}$ . Recall that  $\rho'$  ends in  $(Y_i, s')$  and let  $t' > t_{k-1}$  be any time instant such that  $y \triangleq f'(t') \in Y_i$ . Observe that  $r(t') = s'$ , since there cannot be a state change within the same time slice. Let  $f''$  be the witness trajectory given by the property

of  $reach^+(A, X)$ , which starts in  $y$ , reaches  $X$ , and in the intermediate times remains inside  $A = \llbracket s' \rrbracket \setminus V(s')$ . Now, let  $f$  be the trajectory obtained by concatenating the prefix of  $f'$  ending in  $y$  with  $f''$  and  $\tau = \{t_0, t_1, \dots, t_{k-1}, t^*\}$ , with  $t^* = t' + \epsilon$ , where  $\epsilon$  is the duration of  $f''$ . Observe that  $f(t) \in \llbracket s' \rrbracket$ , for all  $t \in (t_{k-1}, t^*)$ . Indeed,  $(t_{k-1}, t') \subseteq (t_{k-1}, t_k)$  and, by hypothesis,  $f'$  lies in  $\llbracket s' \rrbracket$  in latter interval. Moreover,  $f''$  lies in  $A \subseteq \llbracket s' \rrbracket$  in the interval  $(0, \epsilon)$ , hence  $f$  lies in  $A \subseteq \llbracket s' \rrbracket$  in the interval  $(t', t^*)$ . Clearly, the signal of  $f$  is constant, and equal to  $\lambda(s') \cap AP$ , in the entire interval  $(t_{k-1}, t^*)$ , therefore  $\tau$  is a proper time slicing for  $f$ . Let us set  $\rho \triangleq (f, r)$ , where  $r$  is the continuous run of  $f$  that has the following form:

$$r(t) = \begin{cases} r'(t) & \text{if } 0 \leq t \leq t_{k-1} \\ s' & \text{if } t_{k-1} < t < t^* \\ s & \text{if } t = t^*. \end{cases}$$

Trivially,  $\rho$  satisfies Item (c). Moreover,  $\rho$  satisfies Item (b), since  $\rho'$  avoids  $V' = V$  and at all times  $f''$  is either contained in  $A$ , which is disjoint from  $V(s')$ , or in  $X$ , which is disjoint from  $V(s)$  by assumption. Finally, Item (a) holds as well, since  $w_f = w_{f'} \cdot \lambda(s)$ . Indeed, the trajectory  $f''$  entirely lies in  $\llbracket s' \rrbracket$  except for its last point, which belongs to  $\llbracket s \rrbracket$ .

**[Completeness]** Given  $s \in S$ ,  $P \in Patch(\llbracket s \rrbracket)$ ,  $X \subseteq P$ , and  $V$ , let  $y$  be a point from which there is a hybrid run satisfying Items (a)-(c). Among those hybrid runs, let  $\rho = (f, r^c)$  be one that induces a *shortest* discrete trace, and let  $\tau = \{t_i\}_{i=0}^k$  be the corresponding time slicing. Formally,

$$(\rho, \tau) \in \arg \min_{(f, \_) \in HRun(x), \tau \in TS(\sigma_f)} |trc(\sigma_f, \tau)|.$$

Let  $w \triangleq trc(\sigma_f, \tau)$  and  $k(y, s, P, X, V)$  be the length of  $w$ . We prove that  $y \in \exists Denot(s, P, X, V)$  by induction on  $k(y, s, P, X, V)$ .

- Base case [ $k(y, s, P, X, V) = 1$ ]: By the definition of traces and the fact that all trajectories are right-closed, a trace (and hence its trajectory) is left-closed if and only if its length is odd. Therefore, since the length of  $w$  is 1, the trajectory of  $\rho$  is left-closed, and the corresponding time slicing is  $\tau = \{t_0\}$ . By Item (a), the run ends in  $(X, s)$ . From these two observations, follows that  $s$  is initial and belongs to  $S_{sing}$ . Since the only left-closed trajectories with a discrete trace of length one are those with zero duration, we have that  $f$  starts and ends in  $y$ , which implies  $y \in X$ . By Line 1 of Algorithm 1, we have that  $y \in \exists Denot(s, P, X, V)$ , thus, the thesis follows.
- Inductive case [ $k(y, s, P, X, V) > 1$ ]: Let  $w = w' \cdot \alpha$ , with  $\alpha \subseteq \widehat{AP}$  and  $s' \in S$  the state of  $\mathcal{A}_\varphi^{\text{fin}}$  preceding  $s$  in  $r^c$ . Note that  $\alpha = \lambda(s)$  and  $s \notin S_0$ , by Proposition 1(c). We distinguish two cases.

– [ $s \in S_{sing}$ ]: By Proposition 1(a),  $s' \in S_{open}$ . Let  $A' = reach^+(A, X)$ , with  $A = \llbracket s' \rrbracket \setminus V(s')$ . Since  $f$  is well-behaved and lies in  $A'$  in the

last open slice  $(t_{k-1}, t_k)$  of  $\tau$ , there exists a pair  $(Q, Y) \in \text{split}(A', A)$  and  $\epsilon > 0$  such that  $f$  lies in  $Y \subseteq Q$  at all times in  $(t_{k-1}, t_{k-1} + \epsilon]$ . Consider the prefix  $\rho' = (f_{\leq t_{k-1} + \epsilon}, r_{\leq t_{k-1} + \epsilon}^c)$ , clearly  $\rho'$  ends in  $(Y, s')$  and its discrete trace is obtained from  $w$  by removing the last symbol  $\alpha$ . By applying Lemma 13 to  $\rho'$  and  $(Q, s')$ , there exists a hybrid run  $\rho''$  that starts and ends where  $\rho'$  does, passes only once through  $(Q, s')$ , satisfies  $\text{Visited}(\rho'') \subseteq \text{Visited}(\rho')$ , and the induced discrete trace is no longer than the one of  $\rho'$ . Therefore,  $k(y, s', Q, Y, V) < k(y, s, P, X, V)$ . Hence,  $y$  satisfies the inductive hypothesis w.r.t.  $s', Q, Y$ , and  $V$ , as witnessed by  $\rho''$ , and so  $y \in \exists \text{Denot}(s', Q, Y, V)$ . Since  $(s', s) \in \delta$  and  $(Q, Y) \in \text{split}(A', A)$ , the algorithm at Line 9 adds  $y$  to the set **Result**, which is then returned.

- [ $s \in S_{\text{open}}$ ]: By Proposition 1(a),  $s' \in S_{\text{sing}}$ . Let  $A' = \text{reach}^0(A, X)$ , with  $A = \llbracket s' \rrbracket \setminus V(s')$ . Since  $\rho$  ends in  $(X, s)$ , there exists  $(Q, Y) \in \text{split}(A', A)$  such that  $f(t_{k-1}) \in Y$ . Next, consider the prefixes  $f' = f_{\leq t_{k-1}}$ ,  $\rho' = \rho_{\leq t_{k-1}}$ , and  $\tau' = \{t_i\}_{i=0}^{k-1}$ . Clearly,  $\rho'$  ends in  $(\{f(t_{k-1})\}, s')$ .

Let  $V' = V[s \mapsto V(s) \cup \{P\}]$  as in Line 3 of the algorithm. By Items (b) and (c) on  $\rho$  w.r.t.  $y, s, P, X$ , and  $V$ , it holds that  $\rho'$  avoids  $V$  and  $(P, s)$ . Hence,  $\rho'$  avoids  $V'$ . It follows that  $\rho'$  satisfies Items (a)-(c) with respect to  $y, s', Q, Y$ , and  $V'$ . Moreover, the discrete trace  $\text{trc}(\sigma_{f'}, \tau')$  is strictly shorter than  $w$  by construction. We then have that  $k(y, s', Q, Y, V') < k(y, s, P, X, V)$  and, by inductive hypothesis, we obtain that  $y \in \exists \text{Denot}(s', Q, Y, V')$ . Since  $(s', s) \in \delta$  and  $(Q, Y) \in \text{split}(A', A)$ , the algorithm at Line 9 adds  $y$  to **Result**, which is then returned. ■

Semantics $\gamma$	Section	Assumptions	Abstraction	Formula $\varphi$	$H$	$ V $	Complexity
fin	5.1	–	$Fin(\mathcal{P}, H)$	$\varphi_0$	$H_0$	$K^{H_0+1}$	$K^{O(2^{ \varphi } \cdot K_{\max})}$
	5.2	Omnidirectional	$Omni(\mathcal{P})$	$\varphi_0$	$n/a$	$K^3 + K$	$O(K^3 \cdot 2^{ \varphi })$
inf	6.1	Omnidirectional	$Omni(\mathcal{P})$	$\varphi_0$	$n/a$	$K^3 + K$	$O(K^3 \cdot 2^{ \varphi })$
	6.2	Non-recurrent	$Fin(\mathcal{P}_{stay}, H)$	$\varphi_0 \wedge \psi_{stay}$	$2H_0$	$K^{2H_0+1}$	$K^{O(2^{ \varphi } \cdot K_{\max})}$
may/must	7.1	Omnidirectional	$Omni(\mathcal{P}_{brink}^\gamma)$	$\varphi_0 \wedge \psi_{brink}$	$n/a$	$(BK)^3 + BK$	$O((BK)^3 \cdot 2^{ \varphi })$
			$Omni(\mathcal{P})$	$\varphi_0$	$K^3 + K$		
	7.2	Forced motion	$Fin(\mathcal{P}_{brink}^\gamma, H)$	$\varphi_0 \wedge \psi_{brink}$	$2BH_0$	$(BK)^{2BH_0+1}$	$(BK)^{O(2^{ \varphi } \cdot BK_{\max})}$
7.3	Non-recurrent	$Fin(\mathcal{P}_{brink}^\gamma, H)$	$\varphi_0 \wedge \psi_{brink}$	$2BH_0$	$(BK)^{2BH_0+1}$	$(BK)^{O(2^{ \varphi } \cdot BK_{\max})}$	
			$Fin(\mathcal{P}_{stay}, H)$	$\varphi_0 \wedge \psi_{stay}$	$H_0$	$K^{H_0+1}$	

Table 2: Summary of results.  $H$  is the sufficient horizon,  $V$  is the set of nodes of the polyhedral abstraction,  $K$  is the total number of patches across all observables, including the patches of  $Inv$ ;  $K_{\max}$  is the maximum number of patches of any observable;  $H_0 = 2|S|K_{\max}$  is the sufficient horizon for  $\varphi_0$ ;  $B$  is the number of patches in  $[brink]^\gamma$ . Formula  $\psi_p$  stands for  $\mathbf{F}(p \wedge last)$ . Columns  $H$  and  $|V|$  are meant as upper bounds. The complexity column refers to the cost of computing the polyhedral abstraction and the cost of checking the emptiness of the product between the abstraction and the automaton for the formula. Note that the cost of computing the polyhedral abstraction is measured in terms of symbolic operations on convex polyhedra, as discussed in Section 2.2.