

# Model Checking Linear Temporal Properties on Polyhedral Systems

Massimo Benerecetti ✉ 

Università di Napoli Federico II, Italy

Marco Faella ✉ 

Università di Napoli Federico II, Italy

Fabio Mogavero ✉ 

Università di Napoli Federico II, 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  $\varphi$ .

**2012 ACM Subject Classification** Theory of computation → Modal and temporal logics

**Keywords and phrases** Model Checking, Real-Time Systems, LTLf, RTLf

**Digital Object Identifier** 10.4230/LIPIcs.TIME.2024.19

**Funding** PNRR MUR project PE0000013-FAIR and Indam GNCS 2024 project “Certificazione, Monitoraggio, ed Interpretabilità in Sistemi di Intelligenza Artificiale”. The authors are members of the Gruppo Nazionale Calcolo Scientifico-Istituto Nazionale di Alta Matematica (GNCS-INdAM).

## 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 [20]. 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) [13]. Whereas reachability in LHAs is undecidable [14], 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 on finite traces ( $LTL_f$ ), that we call  $RTL_f$  following Reynolds [22]. Compared to  $LTL_f$  (and  $LTL$ ),  $RTL_f$  does not include an explicit next operator, which is commonly omitted when considering continuous time domains, but includes both a strict and non-strict version of the until operator. 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



© Massimo Benerecetti, Marco Faella, and Fabio Mogavero;  
licensed under Creative Commons License CC-BY 4.0

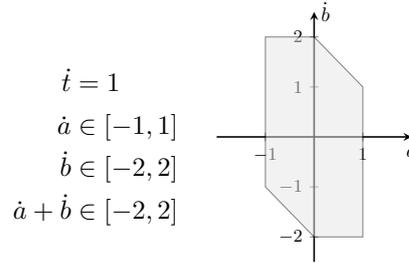
31st International Symposium on Temporal Representation and Reasoning (TIME 2024).

Editors: Pietro Sala, Michael Sioutis, and Fusheng Wang; Article No. 19; pp. 19:1–19:23

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



■ **Figure 1** The flow and its projection on the  $(\dot{a}, \dot{b})$  plane.

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 often within any bounded time interval. We call this notion *well behavedness* and compare it with similar notions in the existing literature.

Our main contribution is a symbolic algorithm to determine the set of initial states from which the system supports a well-behaved trajectory that satisfies a given property, a problem that we call the *existential denotation problem* for  $\text{RTL}_f$ . The algorithm is based on a translation from  $\text{RTL}_f$  to  $\text{LTL}_f$ , followed by the classical automata construction for  $\text{LTL}_f$ . Then, the finite-state automaton is used as a guide for a backward symbolic computation of the existential denotation of the input formula.

The results of the existential denotation problem can be used in two ways, depending on the interpretation given to the input model. Indeed, the non-determinism inherent in a polyhedral 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. 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 with a pump and holding a liquid. An inlet pours liquid into the first tank at an uncertain and time-varying rate, which however is known to be contained in  $[1, 2]$ . The pump shifts liquid from the first tank to the second tank at a varying rate contained in  $[1, 2]$ . Finally, an outlet extracts liquid from the second tank at a varying rate contained in  $[0, 3]$ . 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 the above semantics allows levels to become negative: we guarantee that this does not happen using the formula  $\varphi_{\text{inv}} = \mathbf{G}(a \geq 0 \wedge b \geq 0)$ . Suppose that we want to find the initial states from which the system, within the first 10 time units, can first reach a configuration where  $a \geq b + 1$  and later reach another configuration where  $b \geq a + 1$ . This property is captured by the following formula:

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

This example also shows that, despite not directly supporting time bounds on the temporal operators,  $\text{RTL}_f$  allows to talk about *absolute* time, by introducing an extra variable  $t$  into the model to represent time. In Section 6, we show how our algorithm can readily compute the set of initial points supporting a trajectory that satisfies 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 [16], MITL [2], and STL [18]. Other approaches, including ours, reinterpret the original LTL on real time. In particular, Reynolds investigates the validity problem for LTL interpreted over real time [22].

The dynamics we support generalise the single-mode (i.e., single-location) dynamics of timed automata [1] and constant-rate multi-mode systems (MMS) [4], and correspond to the single-mode dynamics of linear hybrid automata (LHA) [13]. 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 [8]. When it comes to LHAs, even the reachability problem is undecidable [14]. This has not prevented the development of approximate or incomplete approaches, included in tools like SpaceEx [12] and NYCS [6].

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 [15].

**Structure of the Paper.** The paper is organised as follows. Section 2 introduces polyhedral systems and their trajectories, and discusses the notion of well-behavedness and its relationship with other standard regularity conditions. Section 3 defines the classical (i.e., discrete) and continuous semantics of  $\text{LTL}_f$  and  $\text{RTL}_f$ , respectively. Section 4 provides the technical framework to discretise trajectories into traces and  $\text{RTL}_f$  formulae into  $\text{LTL}_f$  formulae. Section 5 presents our algorithm for the existential denotation problem, and Section 6 describes the experiments performed on our prototype implementation.

## 2 Polyhedral Systems, Trajectories, and Signals

We study continuous-time and continuous-state dynamic systems, whose state  $x \in \mathbb{R}^n$  evolves non-deterministically under a differential inclusion of the type  $\dot{x} \in \text{Flow}$ , for a fixed convex polyhedron  $\text{Flow}$ . In the following, we shall use the symbol  $\mathbb{R}^+$  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  $\text{Poly}(\mathbb{R}^n)$  (resp.,  $\text{CPoly}(\mathbb{R}^n)$ ) the set of polyhedra (resp., convex polyhedra) on  $\mathbb{R}^n$ . We shall use the letters  $P, Q$  to refer to convex polyhedra and letters  $A, B, G$  for general polyhedra, instead. For a polyhedron  $G$ , we denote by  $\text{Patch}(G)$  its representation as a finite set of convex polyhedra, called the *patches* of  $G$ . Also,  $\text{cl}(P)$  is *topological closure* of  $P$ , obtained by replacing all strict half-spaces with non-strict ones.

**Atomic propositions.** In the rest of the paper, we assume 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*. That is,  $[p]$  is the set of points where  $p$  holds. 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. That is,

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

By definition,  $\llbracket \alpha \rrbracket$  is a polyhedron. Observe that  $\llbracket \{p\} \rrbracket \subseteq [p]$  and the inclusion may be strict. For instance, if  $[p] = \{x \geq 0\}$  and  $[q] = \{x \geq 2\}$ , then  $\llbracket \{p\} \rrbracket = \{0 \leq x < 2\}$ . Moreover, for any two sets of atomic propositions  $\alpha_1, \alpha_2 \subseteq AP$ , 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.

**Trajectories under polyhedral differential inclusions.** We are interested in dynamic systems that obey a given polyhedral differential inclusion. Therefore, we assume 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 pair  $\mathcal{P} = (Flow, [\cdot])$  a *polyhedral system*.

For a number  $T \in \mathbb{R}^+$ , we use  $\langle 0, T \rangle$  as a shorthand for one of the two right-closed intervals, either  $(0, T]$  or  $[0, T]$ , with left endpoint 0 and right endpoint  $T$ . Given a point  $x \in \mathbb{R}^n$ , a *finite-time trajectory* (*trajectory* from now on) starting from  $x$  is a function  $f: \langle 0, T \rangle \rightarrow \mathbb{R}^n$ , such that: (i)  $\lim_{t \rightarrow 0} f(t) = x$ , (ii)  $f$  is continuous, (iii)  $f$  is differentiable everywhere in its domain except for a finite number of points, (iv) whenever the derivative  $\dot{f}(t)$  is defined, it holds that  $\dot{f}(t) \in Flow$ . When  $\langle 0, T \rangle = [0, T]$  (*resp.*,  $\langle 0, T \rangle = (0, T]$ ) we say that  $f$  is *left-closed* (*resp.*, *left-open*). We use  $Traj(x)$  to denote the set of all trajectories starting from  $x$ .

The interpretation  $[\cdot]$  of the atomic propositions induces a mapping from trajectories to functions of type  $\langle 0, T \rangle \rightarrow 2^{AP}$ , called *bounded signals* [17] (*signals* from now on), 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  $t$  the set of atomic propositions that are true at  $f(t)$ . Formally,

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

For a signal  $\sigma$  and a time  $t \in \langle 0, T \rangle$ , we denote by  $\sigma_{\sim t}$ , with  $\sim \in \{>, \geq\}$ , the left-open or left-closed *suffix of  $\sigma$  starting at  $t$*  defined as follows:  $\sigma_{\sim t}(t') = \sigma(t + t')$ , for all  $t'$  such that  $t' \sim 0$  and  $t + t' \in \langle 0, T \rangle$ .

## 2.1 Well-Behavedness and Finite Variability

A *well-behaved* trajectory  $f: \langle 0, T \rangle \rightarrow \mathbb{R}^n$  is a trajectory that crosses any hyperplane a finite number of times, *i.e.*, for all hyperplanes  $H$  there is a finite set of times  $0 = t_0 < t_1 < \dots < t_k = T$  such that, during every open interval  $(t_i, t_{i+1})$ , the trajectory  $f$  lies in the same closed half-space induced by  $H$ . We denote by  $Traj_{wb}(x)$  the set of all well-behaved trajectories starting from a point  $x \in \mathbb{R}^n$ .

Considering the membership in a half-space as an observable, the condition above 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 [9].

These notions can be compared to classical notion 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 of its domain except for a finite number.<sup>1</sup> The following result follows from Theorem 16 of [9].

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

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

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

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

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

Next, we provide an alternative characterisation of well-behavedness. 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  $t \in \mathbb{R}^+$ , the trajectory changes polyhedron a finite number of times during  $[0, t]$ .*

We say that a signal  $\sigma: \langle 0, T \rangle \rightarrow 2^{AP}$  has *finite variability* if it changes its value only a finite number of times. Formally, this means that there exists a strictly-increasing finite sequence of time points  $0 = t_0 < \dots < t_k = T$  and a finite sequence of observables  $\{\alpha_i\}_{i=0}^{k-1} \subseteq 2^{AP}$  such that, for all indexes  $0 \leq i < k$  and time instants  $t \in (t_i, t_{i+1})$ , it holds true that  $\sigma(t) = \alpha_i$ . We call any such sequence of time points  $\tau = \{t_i\}_{i=0}^k \subseteq \mathbb{R}^+$  a *time-slicing* of  $\sigma$  and denote the set of these sequences  $TS(\sigma)$ . Note that this set does not depend on whether the signal is left-open or not, *i.e.*,  $TS(\sigma) = TS(\sigma_{>0})$ .

As we shall show in the Section 4, the notion of (finite) time-slicing is an essential component of the solution technique proposed in this paper, which reduces the problem of checking  $\text{RTL}_f$  formulae to that of checking  $\text{LTL}_f$  formulae interpreted on finite discrete abstractions of bounded signals. The existence of a time-slicing for a signal as defined above, however, relies on the finite variability property of that signal, as infinite-variability bounded signals do not admit finite 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.*

In the rest of the paper we shall leave the polyhedral system implicit, consider only well-behaved trajectories and, therefore, only finite variability signals. The following table summarises the main semantic notions and their intuitive meaning.

<sup>1</sup> Note that this is a slight adaptation of the classical notion to the case of functions defined on a bounded domain.

■ **Table 1** Main notions used in the paper: three types of trace-like objects (from the most concrete to the most abstract), three types of runs of an automaton, and the time decomposition of a signal.

Type	Name	Role	Symbol
$\langle 0, T \rangle \rightarrow \mathbb{R}^n$	Trajectory	Behaviour of a polyhedral system	$f$
$\langle 0, T \rangle \rightarrow 2^{AP}$	Signal	Interpretation of $\text{RTL}_f$	$\sigma$
$\{0, 1, \dots, k\} \rightarrow 2^{AP}$	Trace	Interpretation of $\text{LTL}_f$	$w$
$\{0, 1, \dots, k\} \rightarrow S$	Discrete run	Behaviour of a finite automaton	$r^d$
$\langle 0, T \rangle \rightarrow S$	Continuous run	Continuous behaviour of a finite automaton	$r^c$
$\langle 0, T \rangle \rightarrow (\mathbb{R}^n \times S)$	Hybrid run	Pairing of a trajectory and a continuous run	$\rho$
$\{0, 1, \dots, k\} \rightarrow \mathbb{R}^+$	Time-slicing	Time decomposition of a signal to generate traces	$\tau$

### 3 Linear Temporal Logics

*Linear Temporal Logic* (LTL) was introduced by Pnueli to specify and verify properties of reactive systems [19]. Given a set of atomic propositions  $AP$ , an LTL formula is composed of atomic propositions, the Boolean connectives *conjunction* ( $\wedge$ ) and *negation* ( $\neg$ ), and the temporal operators *next* ( $X$ ) and two flavors of *until*: strict ( $\dot{U}$ ) and non-strict ( $U$ ).

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

$$\varphi := p \mid \neg \varphi \mid \varphi \wedge \varphi \mid X\varphi \mid \varphi U \varphi \mid \varphi \dot{U} \varphi,$$

where  $p$  is an atomic proposition in  $AP$ . We denote by  $|\varphi|$  the length of formula  $\varphi$ .

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 [22]. In Section 5, 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. Both for the discrete and the continuous version, we consider the bounded semantic fragments, where formulae are interpreted over finite words and bounded signals, respectively.

**Discrete Semantics.** In this paper, we consider the semantic fragment  $\text{LTL}_f$  [10], where formulae are interpreted over non-empty finite words  $w = w_0w_1 \dots w_n$  of symbols in the alphabet  $\Sigma = 2^{AP}$ . For all  $i = 0, \dots, n$ , we denote by  $w_{\geq i}$  the suffix of  $w$  starting from  $w_i$ . The satisfaction relation  $w \models \varphi$  is defined as follows:

- $w \models \varphi$ , for  $\varphi \in AP$ , if and only if  $\varphi \in w_0$ ;
- $w \models \neg \varphi$  if and only if  $w \not\models \varphi$  does not hold;
- $w \models \varphi_1 \wedge \varphi_2$  if and only if  $w \models \varphi_1$  and  $w \models \varphi_2$ ;
- $w \models X\varphi$  if and only if  $|w| > 1$  and  $w_{\geq 1} \models \varphi$ ;
- $w \models \varphi_1 U \varphi_2$  if and only if there exists  $i \geq 0$  such that  $w_{\geq i} \models \varphi_2$  and, for all  $j$  such that  $0 \leq j < i$ , it holds  $w_{\geq j} \models \varphi_1$ ;
- $w \models \varphi_1 \dot{U} \varphi_2$  if and only if there exists  $i > 0$  such that  $w_{\geq i} \models \varphi_2$  and, for all  $j$  such that  $0 < j < i$ , it holds  $w_{\geq j} \models \varphi_1$ .

► **Theorem 5** ([10]). *For all  $\text{LTL}_f$  formulae  $\varphi$  there exists a finite automaton  $\mathcal{A}_\varphi$  that accepts all and only the models of  $\varphi$ .*

**Continuous Semantics.** As it is customary,  $\text{RTL}_f$ , the continuous version of  $\text{LTL}_f$ , is identified as the fragment without the next-time operator  $X$ . The semantics of  $\text{RTL}_f$  formulae is given with respect to signals  $\sigma: \langle 0, T \rangle \rightarrow 2^{AP}$  in the following way:

- $\sigma \models \varphi$ , for  $\varphi \in AP$ , if and only if:
  - $\varphi \in \sigma(0)$ , if  $\sigma$  is left-closed, and
  - there exists  $t' \in \langle 0, T \rangle$  such that  $\varphi \in \sigma(t')$ , for all  $t'' \in (0, t')$ , otherwise;
- $\sigma \models \varphi_1 \cup \varphi_2$  if and only if there exists  $t \in \langle 0, T \rangle$  such that  $\sigma_{\geq t} \models \varphi_2$  and  $\sigma_{\geq t'} \models \varphi_1$ , for all  $t' \in \langle 0, T \rangle$  with  $t' < t$ ;
- $\sigma \models \varphi_1 \dot{\cup} \varphi_2$  if and only if there exists  $0 < t \leq T$  such that  $\sigma_{\geq t} \models \varphi_2$  and  $\sigma_{\geq t'} \models \varphi_1$ , for all  $0 < t' < t$ .

While the base case for left-closed signals is standard, we stipulate that a left-open signal satisfies an atomic proposition  $p \in AP$  if there exists an initial left-open interval contained in the domain of the signal, where  $p$  is observed.

Note that on a left-open signal the semantics of the operators  $\cup$  and  $\dot{\cup}$  coincide. Moreover, unlike in  $\text{LTL}_f$  where the operators  $\dot{\cup}$  and  $\cup$  are inter-derivable thanks to the presence of the operator  $X$ , in  $\text{RTL}_f$  this is not the case and  $\dot{\cup}$  turns out to be strictly more expressive than  $\cup$  (a proof of this result can be found in [21]). Indeed, in both  $\text{LTL}_f$  and  $\text{RTL}_f$ , we have that  $\varphi_1 \cup \varphi_2 \equiv \varphi_2 \vee (\varphi_1 \wedge \varphi_1 \dot{\cup} \varphi_2)$  and in  $\text{LTL}_f$  only it holds, in addition, that  $\varphi_1 \dot{\cup} \varphi_2 \equiv X(\varphi_1 \cup \varphi_2)$ . The semantics of  $\text{RTL}_f$  essentially corresponds to a bounded version of the logic by the same name from [22], except that we consider both left-open and left-closed signals and we omit the past operator *Since*.

**The Problem.** In this work we are interested in solving the problem of computing the *existential denotation* of an  $\text{RTL}_f$  formula defined as follows.

► **Definition 6.** *Given an  $\text{RTL}_f$  formula  $\varphi$  and a polyhedral system  $\mathcal{P}$  on the same set of atomic propositions, the existential denotation of  $\varphi$  on  $\mathcal{P}$  is the set of points of  $\mathbb{R}^n$  from which there exists a well-behaved trajectory whose signal satisfies  $\varphi$ .*

Note that a solution to the existential denotation problem also allows us to solve the *model-checking problem*, where we ask whether a given point  $x \in \mathbb{R}^n$  is the source of some trajectory in  $\mathcal{P}$  whose signal satisfies the formula.

## 4 Discretisation

To address the model-checking problem for  $\text{RTL}_f$ , we reduce it to a suitable decision problem for the discrete version of the logic. Specifically, we show that, for all  $\text{RTL}_f$  formulae  $\varphi$  on a set of atomic propositions  $AP$ , there exists an  $\text{LTL}_f$  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 11. First, we 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  $\tau = \{t_i\}_{i=0}^k \in TS(\sigma)$  of a signal  $\sigma$  decomposes  $\sigma$  into a finite sequence of slices corresponding to an alternation of singular and open time intervals. Introduce the function  $\text{slice}_\tau^\sigma: [0, t_k] \rightarrow \{0, \dots, 2k\}$ , associating each time instant  $t \in [0, t_k]$  with the index of its slice  $\text{slice}_\tau^\sigma(t)$ . Formally:

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

Given a time-slicing  $\tau$  of a signal  $\sigma$ , we now define the discrete trace  $\text{trace}(\sigma, \tau)$  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 (*i.e.*, singular intervals)  $t_i$  of the time-slicing  $\tau$ . Denote again by  $\alpha_i$  the set of observables holding true in the open interval  $(t_i, t_{i+1})$ . The discretisation  $\text{trace}(\sigma, \tau)$  is the finite word defined below for both left-closed and left-open signals. We use  $\text{trace}(\sigma, \tau)_i \subseteq AP \cup \{\text{sing}\}$  to denote the  $i$ -th symbol of the discrete trace. Formally, for a left-closed signal  $\sigma: [0, T] \rightarrow 2^{AP}$  and an index  $j \in \{0, \dots, 2k\}$ , we set:

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

For a left-open signal  $\sigma: (0, T] \rightarrow 2^{AP}$  and an index  $j \in \{0, \dots, 2k-1\}$ , we set:

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

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  $t \leq T$ , we define  $(\{t_i\}_{i=0}^k)_{\geq t} \triangleq \{t'_i\}_{i=0}^{k'}$ , with  $k' \triangleq k-l$ ,  $t'_0 \triangleq 0$ , and  $t'_i \triangleq t_{i+l} - t$ , for all  $i \in \{1, \dots, k'\}$ , where  $l \in \{0, 1, \dots, k\}$  is the maximum index such that  $t_l \leq t$ , which also corresponds to  $\lfloor \frac{\text{slice}_\sigma^\tau(t)}{2} \rfloor$ . Note that, if  $\tau \in TS(\sigma)$ , then  $\tau_{\geq t} \in TS(\sigma_{\geq t}) = TS(\sigma_{>t})$ .

► **Lemma 7.** *Let  $\sigma: \langle 0, T \rangle \rightarrow 2^{AP}$  be a signal,  $\tau \in TS(\sigma)$  one of its time-slicings,  $t \in \langle 0, T \rangle$  a time instant in the signal domain, and  $h = \text{slice}_\sigma^\tau(t)$  the corresponding slice index. Then, it holds true that:*

- $\text{trace}(\sigma, \tau)_{\geq h} = \text{trace}(\sigma_{\geq t}, \tau_{\geq t})$ , if  $\text{sing} \in \text{trace}(\sigma, \tau)_h$ ;
- $\text{trace}(\sigma, \tau)_{\geq h} = \text{trace}(\sigma_{>t}, \tau_{\geq t})$ , otherwise.

In addition, it is immediate to see that a trace of a signal satisfies the following property concerning the auxiliary *sing* atomic proposition. In words, (a) singular and open intervals alternate throughout the trace, (b) the trace must end in a singular interval, and (c) the trace starts in a singular interval *iff* the underlying signal is left-closed.

► **Proposition 8.** *For a signal  $\sigma: \langle 0, T \rangle \rightarrow 2^{AP}$  and a time-slicing  $\tau \in TS(\sigma)$ , it holds that  $\text{trace}(\sigma, \tau) \models \mathbf{G}((\text{sing} \leftrightarrow \mathbf{X}\neg\text{sing}) \vee \text{last}) \wedge \mathbf{F}(\text{last} \wedge \text{sing})$ , where  $\text{last} \triangleq \neg\mathbf{X}\top$ . Moreover,  $\sigma$  is left-closed *iff*  $\text{trace}(\sigma, \tau) \models \text{sing}$ .*

## 4.2 Discretising Formulae

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

$$\begin{aligned}
\text{dsc}(p) &\triangleq p \\
\text{dsc}(\neg\varphi) &\triangleq \neg\text{dsc}(\varphi), \\
\text{dsc}(\varphi_1 \wedge \varphi_2) &\triangleq \text{dsc}(\varphi_1) \wedge \text{dsc}(\varphi_2), \\
\text{dsc}(\varphi_1 \cup \varphi_2) &\triangleq \text{dsc}(\varphi_1) \cup (\text{dsc}(\varphi_2) \wedge (\text{dsc}(\varphi_1) \vee \text{sing})), \\
\text{dsc}(\varphi_1 \dot{\cup} \varphi_2) &\triangleq (\text{sing} \wedge \text{X dsc}(\varphi_1 \cup \varphi_2)) \vee (\neg\text{sing} \wedge \text{dsc}(\varphi_1 \cup \varphi_2)).
\end{aligned}$$

To prove the correctness of the above transformation, we first need to state two properties enjoyed by the semantics of  $\text{RTL}_f$ . In the following, we say that a signal  $\sigma: \langle 0, T \rangle \rightarrow 2^{AP}$  is  $B$ -uniform, for an interval  $B \subseteq \langle 0, T \rangle$ , if  $\sigma(t) = \sigma(t')$ , for all  $t, t' \in B$ .

► **Lemma 9.** *For all  $\text{RTL}_f$  formulae  $\varphi$ , signals  $\sigma: \langle 0, T \rangle \rightarrow 2^{AP}$ , and open intervals  $B \subseteq \langle 0, T \rangle$  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  $\text{RTL}_f$  formula  $\varphi$ .

- **[Base case  $\varphi = p \in AP$ ].** *W.l.o.g.*, let us assume  $\sigma_{\sim_1 t_1} \models \varphi$ . 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  $p \in \sigma_{\geq t_1}(0) = \sigma(t_1)$ . If,  $\sim_1 = >$ , instead, again by the semantics of atomic propositions, this time for left-open signals, there exists a non-empty open interval  $(t_1, t') \subseteq (t_1, T]$  such that  $p \in \sigma_{> t_1}(t' - t_1) = \sigma(t')$ , 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  $C \triangleq [t_2, \text{sup}(B))$  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].** The Boolean operators  $\neg$  and  $\wedge$  are trivial to deal with, so we focus on the strict until operator only, *i.e.*, we consider the case  $\varphi = \varphi_1 \dot{\cup} \varphi_2$ . *W.l.o.g.*, let us assume  $\sigma_{\sim_1 t_1} \models \varphi$ . Independently from the relation  $\sim_1$ , by definition of the semantics of the temporal operator  $\dot{\cup}$ , there exists  $t \in (t_1, T]$  such that  $\sigma_{\geq t} \models \varphi_2$  and  $\sigma_{\geq t'} \models \varphi_1$ , 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  $\sigma_{\geq t'} \models \varphi_1$ . So, 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.
  - **[ $t < \text{sup}(B)$ ].** Since  $\sigma_{\geq t} \models \varphi_2$ , 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  $t \in (t_2, \text{sup}(B))$  satisfies  $\sigma_{\geq t} \models \varphi_2$  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  $\sim_2$ .
  - **[ $t \geq \text{sup}(B)$ ].** Since  $t_2 \in B$ , it holds that  $t_2 < t$ . Hence, to prove that  $\sigma_{\sim_2 t_2} \models \varphi$ , it only remains to show that  $\sigma_{\geq t'} \models \varphi_1$ , for all  $t' \in (t_2, t)$ . Obviously, the open interval  $(t_2, t)$  can be decomposed into the disjoint union of the two adjacent intervals  $(t_2, \text{sup}(B))$  and  $[\text{sup}(B), t)$ . At this point, the required property clearly follows from the fact that  $(t_2, \text{sup}(B)) \subset B$  and  $[\text{sup}(B), t) \subset (t_1, t)$ , as  $\varphi_1$  holds true on all points of these two intervals. ◀

► **Lemma 10.** *For all  $\text{RTL}_f$  formulae  $\varphi$ , signals  $\sigma: \langle 0, T \rangle \rightarrow 2^{AP}$ , and time instants  $t \in \langle 0, T \rangle$ , the following holds true:  $\sigma_{> t} \models \varphi$  iff there exists a time instant  $t' \in (t, T]$  such that  $\sigma_{\geq t'} \models \varphi$ , for all  $t'' \in (t, t')$ .*

The following theorem, which leverages the above two lemmas, establishes the correctness of the discretisation and allows us in the next section to reduce verification of  $\text{RTL}_f$  properties against signals to verification of  $\text{LTL}_f$  properties against discrete traces.

► **Theorem 11.** *For all  $\text{RTL}_f$  formulae  $\varphi$ , signals  $\sigma$ , and time-slicings  $\tau \in \text{TS}(\sigma)$ , it holds that  $\sigma \models \varphi$  iff  $\text{trace}(\sigma, \tau) \models \text{dsc}(\varphi)$ .*

**Proof.** The proof proceeds by structural induction on the formula, where we consider an arbitrary time-slicing  $\tau = \{t_i\}_{0 \leq i \leq k}$  of  $\sigma$ .

- **[Base case  $\varphi = p \in \text{AP}$ ].** For the base case, we distinguish the two cases of left-closed and left-open signals. If  $\sigma$  is left-closed, then, by definition of  $\text{trace}(\sigma, \tau)$ , it holds that  $\text{trace}(\sigma, \tau)_0 = \sigma(0) \cup \{\text{sing}\}$ . Hence, being  $\text{dsc}(p) = p$ , we have  $\sigma \models p$  iff  $\text{trace}(\sigma, \tau) \models \text{dsc}(p)$ . If, on the other hand,  $\sigma$  is left-open, then  $\text{trace}(\sigma, \tau)_0 = \sigma(t)$ , for every  $t \in (0, t_1)$ , 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  $t \in (0, t_1)$ . It immediately follows, then, that  $\sigma \models p$  iff  $\text{trace}(\sigma, \tau) \models \text{dsc}(p)$ .
- **[Inductive case].** We shall focus on the inductive case where  $\varphi = \varphi_1 \dot{\cup} \varphi_2$ , since the cases of the Boolean operators are trivial, while the case for  $\dot{\cup}$  is essentially a simplified version of  $\dot{\cup}$ . In the following, let  $\zeta \triangleq \text{dsc}(\varphi_1 \dot{\cup} \varphi_2)$ .

For the first direction of the equivalence, let us consider the case of a left-closed signal  $\sigma: [0, T] \rightarrow 2^{AP}$  and assume  $\sigma \models \varphi_1 \dot{\cup} \varphi_2$ . Then, by the semantics, there is a  $\bar{t} \in (0, T]$  with  $\sigma_{\geq \bar{t}} \models \varphi_2$  and, for all  $t' \in (0, \bar{t})$ , it holds  $\sigma_{\geq t'} \models \varphi_1$ . Being  $\sigma$  left-closed, it holds that  $\text{sing} \in \text{trace}(\sigma, \tau)_0$ , hence, we only need to show that  $\text{trace}(\sigma, \tau)_{\geq 0} \models \mathbf{X}\zeta$ , i.e.,  $\text{trace}(\sigma, \tau)_{\geq 1} \models \text{dsc}(\varphi_1) \dot{\cup} (\text{dsc}(\varphi_2) \wedge (\text{dsc}(\varphi_1) \vee \text{sing}))$ , by definition of  $\text{dsc}(\varphi_1 \dot{\cup} \varphi_2)$ . We have two more cases, depending on whether  $\bar{t}$  belongs to the time-slicing  $\tau$  or is contained in one of its open intervals. If  $\text{slice}_\sigma^\tau(\bar{t})$  belongs to  $\{t_i\}_{0 \leq i \leq k}$ , let  $\bar{j}$  be the position in the discrete trace corresponding to the instant  $\bar{t}$ , i.e.,  $\bar{j} \triangleq \text{slice}_\sigma^\tau(\bar{t}) > 0$ . Then,  $\text{sing} \in \text{trace}(\sigma, \tau)_{\bar{j}}$  and  $\text{trace}(\sigma_{\geq \bar{t}}, \tau_{\geq \bar{t}}) = \text{trace}(\sigma, \tau)_{\geq \bar{j}}$ , by Lemma 7. By the inductive hypothesis,  $\text{trace}(\sigma_{\geq \bar{t}}, \tau_{\geq \bar{t}}) \models \text{dsc}(\varphi_2)$  and, hence, we obtain  $\text{trace}(\sigma, \tau)_{\geq \bar{j}} \models \text{dsc}(\varphi_2) \wedge \text{sing}$ . If, on the other hand,  $\bar{t} \in (t_i, t_{i+1})$ , for some index  $0 \leq i < k$ , then  $\sigma$  is clearly B-uniform, if we take  $B = (t_i, \bar{t}] \subset (t_i, t_{i+1})$ . Hence, by Lemma 9, we have  $\sigma_{\geq t} \models \varphi_2$ , for all  $t \in B$  and, by Lemma 10 and the fact that  $\inf(B) = t_i$ , we conclude  $\sigma_{> t_i} \models \varphi_2$ . Taking  $\bar{j} \triangleq \text{slice}_\sigma^\tau(\bar{t}) = \text{slice}_\sigma^\tau(t_i) + 1$ , we have that  $\text{sing} \notin \text{trace}(\sigma, \tau)_{\bar{j}}$  and  $\text{trace}(\sigma_{> t_i}, \tau_{\geq t_i}) = \text{trace}(\sigma, \tau)_{\geq \bar{j}}$ , by Lemma 7. By applying the inductive hypothesis, we obtain  $\text{trace}(\sigma_{> t_i}, \tau_{\geq t_i}) \models \text{dsc}(\varphi_2)$ . In this case, we know from the assumption that  $\sigma_{\geq t'} \models \varphi_1$ , for all  $t_i < t' < \bar{t}$ . Then, by applying again Lemma 9 and Lemma 10, we obtain that  $\text{trace}(\sigma_{> t_i}, \tau_{\geq t_i}) \models \text{dsc}(\varphi_1)$ . Thus, we can conclude  $\text{trace}(\sigma, \tau)_{\geq \bar{j}} \models \text{dsc}(\varphi_2) \wedge \text{dsc}(\varphi_1)$ . Regardless of the case, we have obtained that  $\text{trace}(\sigma, \tau)_{\geq \bar{j}} \models \text{dsc}(\varphi_2) \wedge (\text{dsc}(\varphi_1) \vee \text{sing})$ . Let us now consider any  $t' \in J$ , where  $J = (0, t_i)$ , if  $\bar{t} = t_i$ , and  $J = (0, t_i]$ , if  $\bar{t} \in (t_i, t_{i+1})$ , for some index  $0 \leq i < k$ . We have two cases, depending on whether  $t' = t_j$  or  $t' \in (t_j, t_{j+1})$ , for some  $0 < j < i$ . By applying to  $t'$  the same reasoning we applied to  $\bar{t}$  above, using the inductive hypothesis and Lemmas 7, 9, and 10, we obtain that  $\text{trace}(\sigma, \tau)_{\geq j} \models \text{dsc}(\varphi_1)$ , for  $j = \text{slice}_\sigma^\tau(t')$ . Since, in addition,  $\{\text{slice}_\sigma^\tau(t') \mid t' \in J\} = \{1, \dots, \bar{j} - 1\}$ , we can conclude that  $\text{trace}(\sigma, \tau)_{\geq j} \models \text{dsc}(\varphi_1)$ , for all  $1 \leq j < \bar{j}$ . Together with  $\text{trace}(\sigma, \tau)_{\geq \bar{j}} \models \text{dsc}(\varphi_2) \wedge (\text{dsc}(\varphi_1) \vee \text{sing})$ , this gives us  $\text{trace}(\sigma, \tau)_{\geq 1} \models \text{dsc}(\varphi_1) \dot{\cup} (\text{dsc}(\varphi_2) \wedge (\text{dsc}(\varphi_1) \vee \text{sing}))$ , which, in turn, implies  $\text{trace}(\sigma, \tau) \models \text{sing} \wedge \mathbf{X}(\text{dsc}(\varphi_1) \dot{\cup} \text{dsc}(\varphi_2) \wedge (\text{dsc}(\varphi_1) \vee \text{sing}))$ , since  $\text{sing} \in \text{trace}(\sigma, \tau)_0$  in this case. Hence,  $\text{trace}(\sigma, \tau) \models \text{sing} \wedge \mathbf{X}\zeta$  and, finally,  $\text{trace}(\sigma, \tau) \models \text{dsc}(\varphi_1 \dot{\cup} \varphi_2)$  as required.

For the other direction of the equivalence, assume  $\text{trace}(\sigma, \tau) \models \mathbf{dsc}(\varphi_1 \dot{\cup} \varphi_2)$ . Since we are considering a left-closed signal  $\sigma$ , we have  $\text{sing} \in \text{trace}(\sigma, \tau)_0$  and, therefore,  $\text{trace}(\sigma, \tau) \models (\text{sing} \wedge \mathbf{X}\zeta)$ . As a consequence,  $\text{trace}(\sigma, \tau)_{\geq 1} \models \mathbf{dsc}(\varphi_1) \cup (\mathbf{dsc}(\varphi_2) \wedge (\mathbf{dsc}(\varphi_1) \vee \text{sing}))$ . By the semantics of  $\cup$ , there exists an index  $j \geq 1$  such that  $\text{trace}(\sigma, \tau)_{\geq j} \models (\mathbf{dsc}(\varphi_2) \wedge (\mathbf{dsc}(\varphi_1) \vee \text{sing}))$  and  $\text{trace}(\sigma, \tau)_{\geq z} \models \mathbf{dsc}(\varphi_1)$ , for all  $1 \leq z < j$ . We have two cases, depending on whether  $\text{trace}(\sigma, \tau)_j$  contains the proposition  $\text{sing}$  or not. If  $\text{sing} \in \text{trace}(\sigma, \tau)_j$ , then  $\text{trace}(\sigma, \tau)_{\geq j} \models \mathbf{dsc}(\varphi_2) \wedge \text{sing}$  and  $j = \text{slice}_\sigma^\tau(t_i)$ , for some  $0 < i \leq k$ . By Lemma 7,  $\text{trace}(\sigma, \tau)_{\geq j} = \text{trace}(\sigma_{\geq t_i}, \tau_{\geq t_i})$ . Therefore,  $\text{trace}(\sigma_{\geq t_i}, \tau_{\geq t_i}) \models \mathbf{dsc}(\varphi_2)$ . By the inductive hypothesis, then,  $\sigma_{\geq t_i} \models \varphi_2$ . For the other case,  $\text{sing} \notin \text{trace}(\sigma, \tau)_j$ , hence,  $\text{trace}(\sigma, \tau)_{\geq j} \models \mathbf{dsc}(\varphi_2) \wedge \mathbf{dsc}(\varphi_2)$  and  $j = \text{slice}_\sigma^\tau(t)$ , for all  $t \in (t_i, t_{i+1})$  and some  $0 \leq i < k$ . For all such  $t$ , then, we obtain  $\text{trace}(\sigma, \tau)_{\geq j} = \text{trace}(\sigma_{> t}, \tau_{\geq t})$ , thanks to Lemma 7 and, then, also  $\text{trace}(\sigma_{> t}, \tau_{\geq t}) \models \mathbf{dsc}(\varphi_2) \wedge \mathbf{dsc}(\varphi_2)$ . By the inductive hypothesis, it holds  $\sigma_{> t} \models \varphi_2 \wedge \varphi_2$ , for each such  $t$ . Lemma 9, then, gives us  $\sigma_{\geq t} \models \varphi_2 \wedge \varphi_2$ , for all  $t \in (t_i, t_{i+1})$ . Now, take any  $t' \in J$ , where  $J = (0, t_i)$ , if  $\text{sing} \in \text{trace}(\sigma, \tau)_j$ , and  $J = (0, t_i]$ , otherwise. Clearly,  $\text{slice}_\sigma^\tau(t') \in \{1, \dots, j-1\}$  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  $\text{trace}(\sigma, \tau)_{\geq z} \models \mathbf{dsc}(\varphi_1)$  and, by Lemma 7,  $\text{trace}(\sigma, \tau)_{\geq z} = \text{trace}(\sigma_{\geq t_z}, \tau_{\geq t_z})$ , we conclude  $\text{trace}(\sigma_{\geq t_z}, \tau_{\geq t_z}) \models \mathbf{dsc}(\varphi_1)$  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  $z \triangleq \text{slice}_\sigma^\tau(t') < j$ . Lemma 7 in this case gives us  $\text{trace}(\sigma, \tau)_{\geq z} = \text{trace}(\sigma_{> t'}, \tau_{\geq t'})$ . We know that  $\text{trace}(\sigma, \tau)_{\geq z} \models \mathbf{dsc}(\varphi_1)$ , hence,  $\text{trace}(\sigma_{> t'}, \tau_{\geq t'}) \models \mathbf{dsc}(\varphi_1)$ . By the inductive hypothesis,  $\sigma_{> t'} \models \varphi_1$  and, by Lemma 9, also  $\sigma_{\geq t'} \models \varphi_1$ . Putting everything together, we have shown that there is a time  $t \in (0, T]$  such that  $\sigma_{\geq t} \models \varphi_2$  and  $\sigma_{\geq t'} \models \varphi_1$ , for all  $t' \in (0, t]$ , which coincides with the semantic condition for  $\sigma \models \varphi_1 \dot{\cup} \varphi_2$ .

The proof of the inductive case when the signal  $\sigma$  is left-open is essentially the same, except that the first letter  $\text{trace}(\sigma, \tau)_0$  of  $\text{trace}(\sigma, \tau)$  does not contain  $\text{sing}$ , as it corresponds to the first open interval  $(t_0, t_1)$  of the time-slicing, and that  $\mathbf{dsc}(\varphi_1 \dot{\cup} \varphi_2)$  reduces to  $\neg \text{sing} \wedge \zeta$  in this case.  $\blacktriangleleft$

In conclusion, as an immediate corollary of the above result, we have the following theorem, where with every  $\text{RTL}_f$  formula  $\varphi$  we associate the  $\text{LTL}_f$  formula

$$\widehat{\varphi} \triangleq \mathbf{dsc}(\varphi) \wedge \text{sing} \wedge \mathbf{G}((\text{sing} \leftrightarrow \mathbf{X}\neg \text{sing}) \vee \text{last}) \wedge \mathbf{F}(\text{last} \wedge \text{sing}). \quad (1)$$

► **Theorem 12.** *For all  $\text{RTL}_f$  formulae  $\varphi$ , left-closed signals  $\sigma$ , and time-slicings  $\tau \in \text{TS}(\sigma)$ , it holds that  $\sigma \models \varphi$  iff  $\text{trace}(\sigma, \tau) \models \widehat{\varphi}$ .*

## 5 Model Checking $\text{RTL}_f$ on Polyhedral Systems

In this section, we describe the algorithm that solves the existential denotation problem for  $\text{RTL}_f$  on polyhedral systems. 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.

### 5.1 The Basic Operators

The algorithm presented in Section 5.3 (Algorithm 1) requires a function  $\text{reach}^b(A, B)$  that takes as arguments a possibly non-convex polyhedron  $A$  and a convex polyhedron  $B$ , and identifies the set of points of  $A$  that can reach  $B$  while staying in  $A \cup B$ . The  $b$  superscript

## 19:12 Model Checking Linear Temporal Properties on Polyhedral Systems

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 positive delay. Formally, for all polyhedra  $A$  and convex polyhedra  $B$ :

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

Moreover, we need 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  $\{(P_i, X_i)\}_{i=1, \dots, n}$  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. First, define the *positive pre-flow*  $P_{\swarrow_{>0}}$  of a convex polyhedron  $P$  as the set of points that can reach  $P$  after a positive delay. Formally:

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

Lemma 13 below deals with  $reach^0$ , whereas Lemma 14 provides an algorithm for  $reach^+$ . Their proofs can be found in Appendix A.

► **Lemma 13.** *For all polyhedra  $A$  and convex polyhedra  $B$  the following holds:*

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

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:

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

An algorithm for computing  $RWA^m$  using symbolic operations on polyhedra is presented in [7]. The following lemma formalises the connection between  $RWA^m$  and  $reach^+$ .

► **Lemma 14.** *For all polyhedra  $A$  and convex polyhedra  $B$  the following holds:*

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

As far as the *computational complexity* is concerned, first notice that the implementation of the algorithm is based on symbolic operations on polyhedra, whose complexity is already exponential in the worst case. A loose measure of complexity can be obtained by counting the number of symbolic operations involved.

The operator  $reach^0$  involves a constant number of geometric operations, specifically intersections of polyhedra, closure operations and positive time-elapse [5]. The computation of  $reach^+(A, B)$ , instead, requires at most  $|Patch(A)|$  calls to  $RWA^m$ . An analysis of the algorithm for  $RWA^m$  (see Theorem 3 in [7]) shows that computing  $RWA^m(Y, \overline{Z})$  requires at most  $k \cdot m^{O(m)}$  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  $m^{O(m)}$ . Summarising,  $reach^+(A, B)$  requires at most  $m^{O(m)}$  operations, with  $m$  the number of convex patches of  $A$ , since  $B$  is a single patch, and its output contains at most  $m^{O(m)}$  patches.

## 5.2 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  $\hat{\varphi}$  defined in (1).

Let  $\mathcal{A}_{\hat{\varphi}} = (S, \delta, \lambda, S_0, S_F)$  be the finite automaton corresponding to  $\hat{\varphi}$ , according to Theorem 5. Recall that  $\lambda$  labels each state in  $S$  with a subset of  $\widehat{AP} = AP \cup \{\text{sing}\}$ . For convenience, we write  $\llbracket s \rrbracket$  for  $\llbracket \lambda(s) \rrbracket$  to denote the polyhedron interpreting the set of propositions labelling  $s$ .

We assume w.l.o.g. that  $\mathcal{A}_{\hat{\varphi}}$  satisfies the following properties. Properties (a) and (c) are directly encoded in  $\hat{\varphi}$ , while property (b) is enforced via a simple modification of the automaton.

► **Proposition 15.** *The finite automaton  $\mathcal{A}_{\hat{\varphi}}$  satisfies the following properties:*

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

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

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

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

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

## 5.3 The Algorithm

We now describe the main step in the procedure to solve the existential denotation problem, expressed in pseudo-code as the function  $\exists\text{Denot}(\cdot)$  in Algorithm 1. Theorem 18 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 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.

When  $s$  is an initial state, the result is clearly  $X$  itself (Line 1). Otherwise, 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  $(s', s)$ ,

## 19:14 Model Checking Linear Temporal Properties on Polyhedral Systems

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 ( $type(s) = 0$ ), or lingering in  $A$  for some time, if it is open ( $type(s) = +$ ). 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.

■ **Algorithm 1** Function  $\exists\text{Denot}(s, P, X, V)$ . For simplicity, we omit from the notation two implicit arguments: the finite automaton  $\mathcal{A}_{\hat{\varphi}} = (S, \delta, \lambda, S_0, S_F)$  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  $X$ 
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    $A' \leftarrow \text{reach}^{type(s')}(A, X)$ 
7    $\{(Q_1, Y_1), \dots, (Q_n, Y_n)\} \leftarrow \text{split}(A', A)$ 
8   for  $i = 1, \dots, n$  do
9     Result  $\leftarrow$  Result  $\cup \exists\text{Denot}(s', Q_i, Y_i, V')$ 
10 return Result

```

---

The following lemmas state the characteristic properties of the function  $\exists\text{Denot}$ , namely termination (Lemma 16), and soundness and completeness (Lemma 17).

► **Lemma 16.** *For all convex polyhedra  $P$  and  $X$ , 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  $|S|^{O(m \cdot |S|)} \cdot m^{O(m^2 \cdot |S|)}$  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  $1 + 2 \cdot \sum_{s \in S} |\text{Patch}(\llbracket s \rrbracket)|$ . Let  $\chi = (s_0, P_0), (s_1, P_1), \dots$  be the sequence of first and second arguments in a stack of recursive calls to  $\exists\text{Denot}$ , with  $(s_0, P_0)$  being the bottom of the stack. Recall that by design  $s_i \in S$  and  $P_i$  is one of the patches of  $\llbracket s_i \rrbracket$ . Consider a pair  $(s_i, P_i)$  with  $s_i \in S_{\text{open}}$ . The recursive call issued from a state  $s_i$  at recursion level  $i$  adds the patch  $P_i$  to  $V(s_i)$  (Line 3). From that point on, i.e., at recursion levels  $j > i$ , if state  $s'$  considered at Line 4 is  $s_i$ , the assignment at Line 5 ensures that the patch  $P_i$  is not passed to the next recursive call. Hence, either  $s_j \neq s_i$  or  $P_j \neq P_i$ . Equivalently, the pair  $(s_i, P_i)$ , cannot occur again in the sequence  $\chi$ . By the generality of  $(s_i, P_i)$ , we obtain that the sequence  $\chi$  contains no duplicate pairs whose state is in  $S_{\text{open}}$ . Since states in  $\chi$  strictly alternate between  $S_{\text{open}}$  and  $S_{\text{sing}}$ , this 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  $\text{reach}^+$  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  $1 + 2 \cdot m \cdot |S|$  as shown above. Hence, the overall number of symbolic operations required by algorithm is bounded by  $|S|^{O(m \cdot |S|)} \cdot m^{O(m^2 \cdot |S|)}$ . ◀

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  $s \in S_{sing}$  and  $\rho$  is in  $(X, s)$  at the last instant of time in its domain, or  $s \in S_{open}$  and  $\rho$  resides in  $(X, s)$  for some final open time interval bounded by  $t_k$ . Formally:

- if  $s \in S_{sing}$ , 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  $Visited(\rho)$  the set of pairs  $(P, s)$ , composed of a patch  $P \in Patch(\llbracket s \rrbracket)$  and a state  $s$ , traversed by  $\rho$  at any time. We say that a hybrid run  $\rho$  avoids a pair  $(P, s)$  if  $(P, s) \notin Visited(\rho)$ . 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  $s \in S_{open}$ , at most once.

► **Lemma 17.** *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 describes the initial arguments required by Algorithm 1 to solve the existential denotation problem.

► **Theorem 18.** *For all  $RTL_f$  formulas  $\varphi$  and polyhedral systems  $\mathcal{P}$  on the same set of atomic propositions, let  $\hat{\varphi}$  be the corresponding  $LTL_f$  formula,  $\mathcal{A}_{\hat{\varphi}}$  be the finite automaton for  $\hat{\varphi}$ , and*

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

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 12,  $trace(\sigma_f, \tau) \models \hat{\varphi}$ . By definition of  $\mathcal{A}_{\hat{\varphi}}$ , there exists an accepting run  $r \in Runs(\mathcal{A}_{\hat{\varphi}})$  for  $trace(\sigma_f, \tau)$  that ends in some final state  $s \in S_F$ . Let  $\alpha$  be the last symbol of  $trace(\sigma_f, \tau)$ , 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 17 (conditions (b) and (c) hold trivially for  $\rho$ ). Therefore, Lemma 17 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 17, 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  $r \in Runs(\mathcal{A}_{\hat{\varphi}})$  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  $trace(\sigma_f, \tau)$ . This means that  $trace(\sigma_f, \tau) \models \hat{\varphi}$  and Theorem 12, then, ensures that  $\sigma_f \models \varphi$ . ◀

## 6 Experiments

In this section, we report on the experiments performed with our implementation, which is based on Parma Polyhedral Library [5] as the underlying engine for the symbolic manipulation of polyhedra. Our prototype implementation starts from an  $\text{RTL}_f$  formula  $\varphi$  and computes its discretisation  $\widehat{\varphi}$  according to Equation (1). This formula is translated into standard LTL (see [10]) in order to obtain a non-deterministic Büchi automaton recognising its models using SPOT [11]. The NBA is then turned into an NFA  $\mathcal{A}$  recognising the finite traces satisfying  $\widehat{\varphi}$ . 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).

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:

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

where  $k \geq 1$ ,  $\star \in \{\text{gap}, \text{nogap}\}$ , and the interpretations of the atomic propositions is reported in the following table:

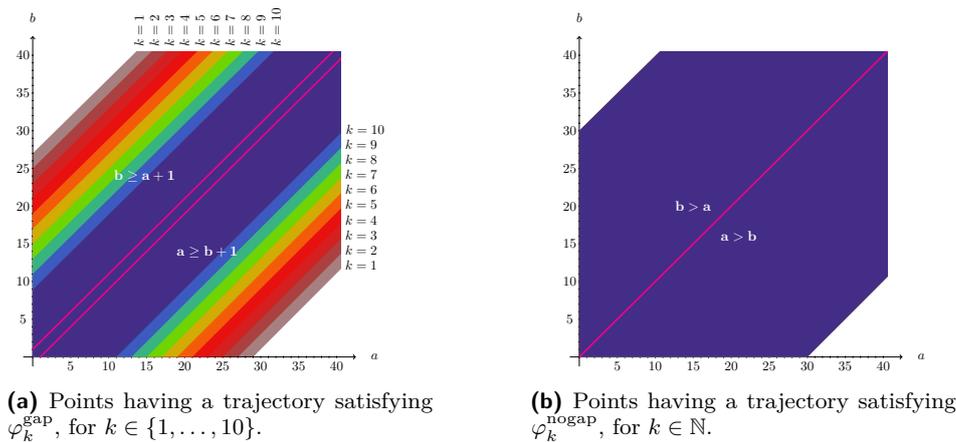
	$[p]$	$[q]$	$[\text{inv}]$	$[t_0]$	$[t_{\max}]$
$\varphi_k^{\text{gap}}$	$a \geq b + 1$	$b \geq a + 1$	$a \geq 0 \wedge b \geq 0$	$t = 0$	$t \leq 10$
$\varphi_k^{\text{nogap}}$	$a > b$	$b > a$	$a \geq 0 \wedge b \geq 0$	$t = 0$	$t \leq 10$

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 disjoint 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 2 shows the denotations of the two families of formulas, both limited to the cross section for  $t = 0$ . In particular, Figure 2a 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 2b,



■ **Figure 2** The results of the experiments for the two families of  $\text{RTL}_f$  properties.

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  $k$ .

## References

- 1 R. Alur and D.L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126:183–235, 1994. doi:10.1016/0304-3975(94)90010-8.
- 2 R. Alur, T. Feder, and T.A. Henzinger. The benefits of relaxing punctuality. *Journal of the ACM (JACM)*, 43(1):116–146, 1996. doi:10.1145/227595.227602.
- 3 R. Alur, T.A. Henzinger, and P.-H. Ho. Automatic symbolic verification of embedded systems. *IEEE Trans. Softw. Eng.*, 22:181–201, March 1996. doi:10.1109/32.489079.
- 4 R. Alur, A. Trivedi, and D. Wojtczak. Optimal scheduling for constant-rate multi-mode systems. In *Proceedings of the 15th ACM international conference on Hybrid Systems: Computation and Control*, pages 75–84, 2012.
- 5 R. Bagnara, P. M. Hill, and 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):3–21, 2008. doi:10.1016/j.scico.2007.08.001.
- 6 M. Benerecetti and M. Faella. Automatic synthesis of switching controllers for linear hybrid systems: Reachability control. *ACM Trans. on Embedded Computing Systems*, 16(4), 2017. doi:10.1145/3047500.
- 7 M. Benerecetti, M. Faella, and S. Minopoli. Automatic synthesis of switching controllers for linear hybrid systems: Safety control. *Theoretical Computer Science*, 493:116–138, 2013. doi:10.1016/J.TCS.2012.10.042.
- 8 M. Blondin, P. Offtermatt, and A. Sansfaçon-Buchanan. Verifying linear temporal specifications of constant-rate multi-mode systems. In *LICS*, pages 1–13, 2023. doi:10.1109/LICS56636.2023.10175721.
- 9 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, pages 47–58. Morgan Kaufmann, 1992.
- 10 G. De Giacomo and M.Y. Vardi. Linear temporal logic and linear dynamic logic on finite traces. In Francesca Rossi, editor, *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*, pages 854–860. IJCAI/AAAI, 2013. URL: <http://www.aaai.org/ocs/index.php/IJCAI/IJCAI13/paper/view/6997>.

- 11 A. Duret-Lutz, A. Lewkowicz, A. Fauchille, T. Michaud, E. Renault, and L. Xu. Spot 2.0—a framework for ltl and-automata manipulation. In *International Symposium on Automated Technology for Verification and Analysis*, pages 122–129. Springer, 2016.
- 12 G. Frehse, C. Le Guernic, A. Donzé, S. Cotton, R. Ray, O. Lebeltel, R. Ripado, A. Girard, T. Dang, and O. Maler. Spaceex: Scalable verification of hybrid systems. In *CAV 11: Proc. of 23rd Conf. on Computer Aided Verification*, pages 379–395, 2011. doi:10.1007/978-3-642-22110-1\_30.
- 13 T.A. Henzinger. The theory of hybrid automata. In *11th IEEE Symp. Logic in Comp. Sci.*, pages 278–292, 1996. doi:0.1109/LICS.1996.561342.
- 14 T.A. Henzinger, P.W. Kopke, A. Puri, and P. Varaiya. What’s decidable about hybrid automata? *J. of Computer and System Sciences*, 57(1):94–124, 1998. doi:10.1006/jcss.1998.1581.
- 15 M. Kloetzer and C. Belta. A fully automated framework for control of linear systems from temporal logic specifications. *IEEE Transactions on Automatic Control*, 53(1):287–297, 2008. doi:10.1109/TAC.2007.914952.
- 16 R. Koymans. Specifying real-time properties with metric temporal logic. *Real-time systems*, 2(4):255–299, 1990. doi:10.1007/BF01995674.
- 17 O. Maler and D. Nickovic. Monitoring temporal properties of continuous signals. In *Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems*, pages 152–166. Springer, 2004. doi:10.1007/978-3-540-30206-3\_12.
- 18 O. Maler, D. Nickovic, and 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*, pages 475–505, 2008. doi:10.1007/978-3-540-78127-1\_26.
- 19 A. Pnueli. The temporal logic of programs. In *18th Annual Symposium on Foundations of Computer Science, Providence, Rhode Island, USA, 31 October - 1 November 1977*, pages 46–57. IEEE Computer Society, 1977. doi:10.1109/SFCS.1977.32.
- 20 R. Poovendran. Cyber–physical systems: Close encounters between two parallel worlds. *Proceedings of the IEEE*, 98(8):1363–1366, 2010.
- 21 M. Reynolds. The complexity of the temporal logic with “until” over general linear time. *Journal of Computer and System Sciences*, 66(2):393–426, 2003. doi:10.1016/S0022-0000(03)00005-9.
- 22 M. Reynolds. The complexity of temporal logic over the reals. *Annals of Pure and Applied Logic*, 161(8):1063–1096, 2010. doi:10.1016/J.APAL.2010.01.002.

## A Additional Proofs

► **Proposition 2.** *On the set of 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  $a > 0$ . ◀

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

**Proof.** Take any polyhedral partitioning  $\{P_i\}_{i \in I}$  of  $\mathbb{R}^n$  that respects the propositions in  $AP$ , meaning that, for all  $i \in I$  and  $p \in AP$ , either  $P_i \cap [p] = \emptyset$  or  $P_i \subseteq [p]$ . Since  $f$  is well-behaved, it must change convex polyhedron in  $\{P_i\}_{i \in I}$  a finite number of times in  $\langle 0, T \rangle$ . 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  $f$  changes polyhedron in the sequence, with the possible addition of instants 0 and  $T$ , if needed. Since the polyhedral partitioning respects  $AP$ , every  $P_{i_j}$  is contained in  $\llbracket \alpha \rrbracket$ , for some  $\alpha \subseteq AP$ . Hence,  $\tau$  is a suitable time-slicing of  $f$ . The thesis follows then from the finite length of  $\tau$ .  $\blacktriangleleft$

► **Lemma 10.** For all  $\text{RTL}_f$  formulae  $\varphi$ , signals  $\sigma: \langle 0, T \rangle \rightarrow 2^{AP}$ , and time instants  $t \in \langle 0, T \rangle$ , the following holds true:  $\sigma_{>t} \models \varphi$  iff there exists a time instant  $t' \in (t, T]$  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  $\dot{\cup}$  temporal formulae. Since the inductive cases of Boolean operators  $\neg$  and  $\wedge$  are trivial to deal with, here we focus on the base cases for atomic propositions and  $\dot{\cup}$  only. Recall that  $\dot{\cup}$  is a derived operator.

- **[Base case  $\varphi = p \in AP$ ].** Since  $\sigma_{>t}$  is a left-open signal, by the semantic of atomic propositions,  $\sigma_{>t} \models p$  holds iff there exists a non-empty open interval  $(t, t') \subseteq (t, T]$  such that  $p \in \sigma_{>t}(t'' - t) = \sigma(t'')$ , 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  $\varphi = \varphi_1 \dot{\cup} \varphi_2$ , only-if direction].** By the semantics of the temporal operator  $\dot{\cup}$ , if  $\sigma_{>t} \models \varphi_1 \dot{\cup} \varphi_2$ , then there exists  $t_2 \in (t, T]$  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  $\sigma_{\geq t'} \models \varphi_1 \dot{\cup} \varphi_2$ , for all  $t'' \in (t, t')$ .
- **[Base case  $\varphi = \varphi_1 \dot{\cup} \varphi_2$ , if direction].** Let  $\tau = \{t_i\}_{i=0}^k \in TS(\sigma_{>t})$  be a time-slicing of the suffix  $\sigma_{>t}$  of the signal  $\sigma$  and  $j \in \{1, \dots, k\}$  the smallest index such that either (a)  $\sigma_{\geq t+t_j} \models \varphi_2$  or (b)  $\sigma_{\geq t+t''} \models \varphi_2$ , 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 open interval  $(t, t')$  the until formula  $\varphi_1 \dot{\cup} \varphi_2$  is satisfied. Now, suppose by contradiction that  $\sigma_{>t} \not\models \varphi_1 \dot{\cup} \varphi_2$ . Since  $\varphi_2$  is satisfied either at time instant  $t + t_j$  or at all time instants  $t + t''$ , 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  $t_1$ , either in  $(t, t + t_j)$  or in  $(t, t_{j-1}]$ , such that  $\sigma_{\geq t_1} \not\models \varphi_1$ . However, this would clearly lead to  $\sigma_{\geq t'} \not\models \varphi_1 \dot{\cup} \varphi_2$ , for all  $t'' \in (t, t_1)$ , which contradicts the hypothesis.  $\blacktriangleleft$

The following lemma is a straightforward adaptation of Lemma 1 from [3], used in the proofs of Lemma 13 and Lemma 14.

► **Lemma 19 ([3]).** For any convex flow constraint  $Flow$ , 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  $d \in Flow$ , such that  $f'(0) = x_1$ ,  $f'(t^*) = x_2$  and  $f'(t) \in X$ , for all  $t \in [0, t^*]$ .

► **Lemma 13.** For all polyhedra  $A$  and convex polyhedra  $B$  the following holds:

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

**Proof.** Let  $\Delta \triangleq A \cap cl(B) \cap B_{\angle_{>0}}$ . To show that  $\Delta \subseteq X^0$ , we just need to observe first that if  $x \in \Delta$ , then  $x \in A$ . Moreover,  $x \in B_{\angle_{>0}}$ , which means that there is a direction  $d \in Flow$  such that  $x + d \cdot t \in B$ , for some  $t > 0$ . In addition, since  $x \in cl(B)$  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  $cl(B)$ . Since  $y$  can be reached from  $x$  following an admissible trajectory, by convexity of  $Flow$  and Lemma 19, there is a direction  $d \in Flow$  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  $cl(B)$ . Hence, we conclude that  $x \in A \cap cl(B) \cap B_{\angle_{>0}}$ , as required.  $\blacktriangleleft$

► **Lemma 14.** *For all polyhedra  $A$  and convex polyhedra  $B$  the following holds:*

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

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

$$RWA^m(T_P, \bar{A}) = \{x \in \mathbb{R}^n \mid \exists f \in Traj_{wb}(x), t \geq 0 : f(t) \in T_P \text{ and } \forall t' \in [0, t] : f(t') \in A\}.$$

Consider any point  $x \in RWA^m(T_P, \bar{A})$ , a trajectory  $f$  witnessing its membership to the set  $RWA^m(T_P, \bar{A})$ , 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  $y \in (cl(P) \cap B)_{\angle_{>0}}$ . 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  $f'(t^*) \in cl(P) \cap B$ . 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  $f \in Traj_{wb}(x)$  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  $P \in Patch(A)$ , we have that  $RWA^m(T_P, \bar{A}) \subseteq reach^+(A, B)$ .

For the other direction, assume  $x \in reach^+(A, B)$ . Then there exist  $f \in Traj_{wb}(x)$  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  $Patch(A)$  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  $f(t) \in cl(P) \cap B$ . 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  $Flow$  and thanks to Lemma 19, we obtain that  $f(t') \in P \cap (cl(P) \cap B)_{\angle_{>0}} = T_P$ , 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  $reach^+(A, B) \subseteq \bigcup_{P \in Patch(A)} RWA^m(T_P, \bar{A})$ .  $\blacktriangleleft$

The next lemma is instrumental in proving the completeness of Algorithm 1 in Lemma 17.

► **Lemma 20.** *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 trajectory is feasible. By convexity of  $P$ , such trajectory is 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 17.** *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.*

**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 initial state of  $\mathcal{A}_{\hat{\varphi}}$  is in itself a run of  $\mathcal{A}_{\hat{\varphi}}$  of length 1 from an initial state. If  $s$  is an initial state, by Proposition 15(a) it includes the *sing* proposition. Then, 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  $\mathcal{A}_{\hat{\varphi}}$ . 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  $s \notin S_{open}$ .

Next, we 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  $\mathcal{A}_{\hat{\varphi}}$  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  $s_i \in S_{open}$  then  $P_i \notin V(s_i)$ , (v) all the pairs  $(s_i, P_i)$  such that  $s_i \in S_{open}$  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  $A' \triangleq reach^{type(s')}(A, X)$  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, there must be a pair  $(Q, Y) \in split(A', A)$ , 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  $split(A', A)$ . 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  $s \in S_{open}$ , 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  $s \in S_{open}$ , 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  $s \in S_{sing}$ , 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 *reach*, while satisfying the Items (a), (b), and (c) of the lemma. We again distinguish two cases.

- [ $s' \in S_{sing}$ ] By Proposition 15(c),  $\text{type}(s') = 0$  and  $s \in S_{open}$ . 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  $H\text{Run}(x)$  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  $V' = V[s \mapsto V(s) \cup P]$ .

- [ $s' \in S_{open}$ ] Obviously,  $\text{type}(s') = +$ , and  $s \in S_{sing}$ . 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  $\text{reach}^+(A, X)$ , 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. 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  $s \in S$ ,  $P \in \text{Patch}(\llbracket s \rrbracket)$ ,  $X \subseteq P$ , 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,  $(\rho, \tau) \in \arg \min_{(f, \tau) \in \text{HRun}(x), \tau \in \text{TS}(\sigma_f)} |\text{trace}(\sigma_f, \tau)|$ .

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

- Base case [ $k(y, s, P, X, V) = 1$ ]: By Item (a),  $s$  is an initial state. By Proposition 15(a),  $s \in S_{\text{sing}}$ . 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 Item (a). By Line 1 of Algorithm 1, we have that  $y \in \exists \text{Denot}(s, P, X, V)$ , 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  $\mathcal{A}_{\hat{\varphi}}$  preceding  $s$  in  $r^c$ . Note that  $\alpha = \lambda(s)$  and  $s \notin S_0$ , by Proposition 15(b). We distinguish two cases.

[ $s \in S_{\text{sing}}$ ]: By Proposition 15(c),  $s' \in S_{\text{open}}$ . Let  $A' = \text{reach}^+(A, X)$ , 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  $\rho' = (f_{\leq t+\epsilon}, r^c_{\leq t+\epsilon})$ , clearly  $\rho'$  ends in  $(Y, s')$  and its discrete trace is obtained from  $w$  by removing the last symbol  $\alpha$ . By applying Lemma 20 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 15(c),  $s' \in S_{\text{sing}}$ . 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  $\text{trace}(\sigma_{f'}, \tau')$  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. ◀