Relating Syntactic and Semantic Perturbations of Hybrid Automata

Nima Roohi
Univeristy of Pennsylvania, USA
roohi2@cis.upenn.edu

Pavithra Prabhakar
Kansas State University, USA
http://people.cs.ksu.edu/~pprabhakar/
pprabhakar@ksu.edu

Mahesh Viswanathan
University of Illinois at Urbana-Champaign, USA
http://vmahesh.cs.illinois.edu/
vmahesh@illinois.edu

Abstract

We investigate how the semantics of a hybrid automaton deviates with respect to syntactic perturbations on the hybrid automaton. We consider syntactic perturbations of a hybrid automaton, wherein the syntactic representations of its elements, namely, initial sets, invariants, guards, and flows, in some logic are perturbed. Our main result establishes a continuity like property that states that small perturbations in the syntax lead to small perturbations in the semantics. More precisely, we show that for every real number $\epsilon > 0$ and natural number k, there is a real number $\delta > 0$ such that $\mathcal{H}^\delta$, the δ syntactic perturbation of a hybrid automaton $\mathcal{H}$, is ϵ-simulation equivalent to $\mathcal{H}$ up to k transition steps. As a byproduct, we obtain a proof that a bounded safety verification tool such as `dReach` will eventually prove the safety of a safe hybrid automaton design (when only non-strict inequalities are used in all constraints) if `dReach` iteratively reduces the syntactic parameter δ that is used in checking approximate satisfiability. This has an immediate application in counter-example validation in a CEGAR framework, namely, when a counter-example is spurious, then we have a complete procedure for deducing the same.

2012 ACM Subject Classification Computer systems organization → Embedded and cyber-physical systems, Theory of computation → Timed and hybrid models, Software and its engineering → Model checking

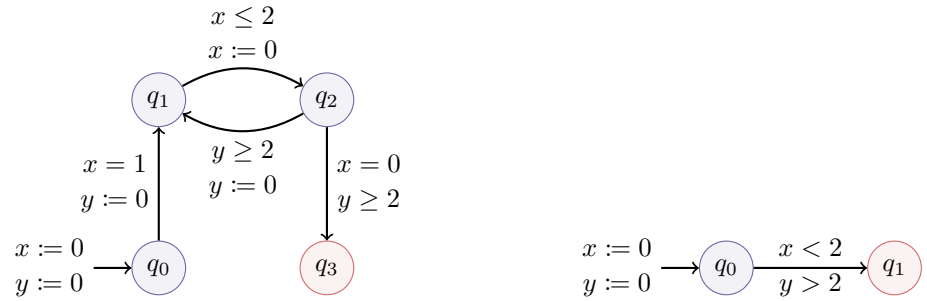
Keywords and phrases Model Checking, Hybrid Automata, Approximation, Perturbation

Digital Object Identifier 10.4230/LIPIcs.CONCUR.2018.26

Funding Mahesh Viswanathan was partially supported by NSF CSR 1422798, and Pavithra Prabhakar was partially supported by NSF CAREER Award No. 1552668 and ONR YIP Award No. N00014-17-1-257.

1 Introduction

Hybrid automata are a mathematical framework to model systems consisting of a digital controller interacting with a continuously evolving physical process. Such cyberphysical systems arise in a variety of applications ranging from every day smart home appliances to safety critical systems like avionics software and self driving cars. Hybrid automata have discrete modes corresponding to phases in the digital controller, where the physical



(a) Timed automaton T_A.

(b) Timed automaton T_B.

Figure 1 Example timed automata whose syntactic perturbations are not semantically close. Continuous variables x, y are clocks, i.e., $\dot{x} = \dot{y} = 1$ and the invariant in each state is $0 \leq x, y \leq 3$. The states q_3 in T_A and state q_1 in T_B are not reachable. But q_1 and q_3 are reachable in infinitesimal syntactic perturbations T_A and T_B, respectively.

environment evolves continuously according to physics laws based on the actuator inputs from the controller in that phase. Transitions between modes model discrete changes to actuator inputs from the controller based on sensor feedback.

Formal models of cyberphysical systems are typically "best effort" descriptions that may not be 100% faithful to the actual system. There are several sources of inaccuracies. Environment parameters, like network latency, are estimated based on extensive experimentation. Differential equations governing the behavior of the physical plant maybe imprecise, either because of limitations in our mathematical understanding of the physics, or because of a conscious effort to construct a tractable model by approximating. Sensor and actuator delays might either be unpredictable or have been ignored. Finally, inaccuracies in sensor input to the controller may not have been faithfully modeled.

For these reasons, a system modeled by hybrid automaton $\mathcal{H}$, may, in practice, behave like the automaton $\mathcal{H}^\delta$ which is obtained from $\mathcal{H}$ by syntactically perturbing constants, constraints on mode switches, and flow equations governing continuous evolution, by some $\delta > 0$. A natural question to ask is if the automaton $\mathcal{H}$ and its perturbation $\mathcal{H}^\delta$ are *semantically close* (in some well defined sense). Can a perturbed automaton $\mathcal{H}^\delta$ be arbitrarily close to $\mathcal{H}$? The challenge in answering this question lies in the presence of discrete mode changes – small changes to the behavior of a hybrid automaton could result in transitions becoming enabled that yield unexpected behavior in the perturbed automaton.

For example, consider the timed automaton T_A shown in Figure 1a. Variables x, y are clocks (i.e., $\dot{x} = \dot{y} = 1$), and the invariant in every location is $0 \leq x, y, \leq 3$. Observe that, when the automaton first visits q_1 from q_0, $x = 1$ and y is set to 0. Because of this, when the automaton switches between q_1 and q_2, it spends *at most* 1 time unit in q_1 and *at least* 1 time unit in q_2. Further since all transitions into q_1 set y to 0, whenever the automaton transitions to q_2, x is set to 0, and y is at most 1. Therefore, the transition to q_3 is never enabled, and q_3 is not reachable. However, perturbing T_A slightly by changing the guard from q_1 to q_2 to $x \leq 2 + \delta$ and the guard from q_2 to q_1 to $y \geq 2 - \delta$ (for some small δ) makes q_3 reachable – in the perturbed automaton, we can ensure that the automaton stays for $1 + n\delta$ units during its n^{th} visit to q_1, and so after visiting q_1 n_* times (for n_* that satisfies $1 + n_*\delta \geq 2$), when we reach q_2, the transition to q_3 will be enabled. Thus, even a small δ-perturbation of T_A, results in an automaton T_A^δ that is not semantically close, as q_3 is reachable in T_A^δ but not reachable in T_A. The difference between T_A and T_A^δ arises when

one considers executions with arbitrarily many transitions. Does the property of semantic closeness hold if one considers executions of bounded number of steps? The answer once again is no. Consider the example timed automaton T_B in Figure 1b. Once again, x, y are clocks and the invariant in every location is $0 \leq x, y \leq 3$. The transition from q_0 to q_1 is not enabled in T_B and so q_1 is not reachable. However, perturbing the guard to $x < 2 + \delta$ and $y > 2 - \delta$, results in an automaton where q_1 is reachable, and hence not semantically close to T_B.

The examples in Figure 1 illustrate that in general syntactic perturbations can result in models that are not semantically close. Any affirmative results, need to account for the subtle issues that arise in the examples of Figure 1. Our main result is that for a fairly general class of hybrid automata, that include hybrid automata with highly non-linear and non-deterministic dynamics, syntactic perturbations are closely related to semantic perturbations. We consider hybrid automata $\mathcal{H}$ all of whose components, like flows and invariants in modes, and guards and resets on transitions, are described using formulas in first-order logic over reals built from constraints of the form $f \geq 0$, where f is a continuous function, and using conjunction, disjunction, and first order quantification. For a formula φ in this logic, its perturbation by δ, φ^δ, is the formula obtained by replacing all atomic constraints $f \geq 0$ in φ by $f + \delta \geq 0$. Using the notion of perturbation of a constraint, we define $\mathcal{H}^\delta$ to be the hybrid automaton obtained from $\mathcal{H}$ by perturbing all constraints φ appearing in $\mathcal{H}$ by δ. We show that for any $\epsilon \in \mathbb{R}_+$ and $k \in \mathbb{N}$, there is a $\delta \in \mathbb{R}_+$ such that $\mathcal{H}^\delta$ is ϵ-simulated for k-steps by $\mathcal{H}$. In other words, every execution ρ of $\mathcal{H}^\delta$ (having at most k discrete transitions) is simulated by an execution ρ' of $\mathcal{H}$ such that the states of ρ and ρ' are within distance ϵ at all times. Our definition of perturbation ensures that $\mathcal{H}$ is always simulated (in the formal sense) by $\mathcal{H}^\delta$. Therefore, we show that $\mathcal{H}$ and $\mathcal{H}^\delta$ are approximately simulation equivalent for k steps. Thus, one way to informally interpret our results is as follows. Let us consider the function $[\![\mathcal{H}]\!]$ that maps an automaton to its transition system semantics. Our results can be seen as saying that $[\![\cdot]\!]$ is a "continuous" map with the metric on the hybrid automata induced by δ perturbations.

The crux of our result is a technical lemma that maybe of independent interest. Consider any formula φ in the logic defined in the previous paragraph. Let us assume that all free variables are constrained to take values from a bounded interval $\mathbb{I}$. If X is the set of free variables of φ, then we can see φ as defining a subset (denoted $[\![\varphi]\!]$) of $\mathbb{I}^X$ in the standard way. We show that, for any formula φ, and any $\epsilon > 0$, there is a $\delta > 0$ such that the set $[\![\varphi^\delta]\!]$ (φ^δ is the δ-perturbation of φ) is contained in the ϵ-ball around $[\![\varphi]\!]$.

Our results on relating syntactic and semantic perturbations have implications in satisfiability checking as well as verification, that served as our initial motivation for studying this problem. Our first application is in the realm of δ-complete decision procedures [10, 11], that take as input a formula in first order logic and a parameter δ, and return either that the formula is unsatisfiable or that a δ syntactic perturbation of it is satisfiable. Note that in the case that the formula is unsatisfiable, but a δ-perturbation of it is satisfiable, the procedure is allowed to return either of the answers. Our results guarantee that if the formula is unsatisfiable, then there is a δ for which the procedure will return `unsat`, and such a δ can be computed by simply starting from 1 and halving it iteratively. This has direct implications on the bounded safety verification using a tool such as `dReach` [12], that uses a δ-complete decision procedure to check the satisfiability of the formula encoding the bounded verification problem of hybrid automata. More precisely, `dReach` takes as input a hybrid automaton $\mathcal{H}$, a bound on the discrete steps k, an unsafe set U and a δ, and outputs either that $\mathcal{H}$ is safe with respect to U and k discrete steps, or that $\mathcal{H}^\delta$ is unsafe. Our results imply that if

$\mathcal{H}$ is safe, then there is a δ for which `dReach` will necessarily conclude safety, and such a δ again can be computed. Finally, these results are relevant in the context of counter-example guided abstraction refinement (CEGAR) based analysis for hybrid automata [26], wherein the above guarantees on bounded safety verification imply that the validation succeeds if the counter-example is spurious and hence the CEGAR loop makes progress.

Related Work

There has been previous work on relating syntactic perturbation of first order logic formulas over reals to their semantic perturbation [8, 24, 25]. These papers show that small syntactic perturbations result in small semantic perturbations. However, the notion of distance between semantic sets is different from ours. The distance between two subsets A, B of $\mathbb{R}^n$ is taken to be the volume of the symmetric difference between A and B. Thus, in [8, 24, 25], the distance between $[\![\varphi]\!]$ and $[\![\psi]\!]$ may be 0 even if $[\![\varphi]\!] \neq [\![\psi]\!]$. This is not true for us, and is one of the reasons our proofs rely significantly on the Bolzano-Weierstrass theorem. In addition, using the observations in [8, 24, 25], one cannot prove that syntactically perturbing a hybrid automaton results in a model that is ϵ-simulation equivalent.

The results in this paper are closely related to *robustness verification* [1, 3, 4, 7, 9, 14, 15, 23, 27, 29]. In robustness verification, the goal is to develop algorithms to determine if a system $\mathcal{H}$ and all its perturbations $\mathcal{H}^\delta$ in some small neighborhood, satisfy the correctness property φ. While establishing the semantic proximity of $\mathcal{H}$ and $\mathcal{H}^\delta$ may help answer the robustness verification question, verification per se, is not the focus of this paper. Our principal goal is to answer the meta-mathematical question of the relationship between syntactic and semantic perturbations of a hybrid automaton.

Another related line of work consists of methods for constructing simplified models of hybrid systems that are semantically close to the original system [13, 17, 18, 20–22] to reduce the complexity of verification. Semantic closeness is expressed using the notions of approximate simulations and bisimulations which establish that every execution of one system can be matched by a corresponding execution of the other system that stays "close" to it. Approximate bisimulation is weaker than bisimulation, however, it allows for the construction of simplified models that are semantically close for a large class of systems [20] under certain stability assumptions.

2 Preliminaries

2.1 Functions and Sets

The set of *natural*, *positive natural*, *real*, *non-negative real*, and *positive real* numbers are represented by $\mathbb{N}$, $\mathbb{N}_+$, $\mathbb{R}$, $\mathbb{R}_{\geq 0}$, and $\mathbb{R}_+$, respectively. For any two sets A and B, power set of A is denoted by 2^A, the Cartesian product of A and B is denoted by $A \times B$, and the set of functions from A to B is denoted by $A \to B$ or B^A. For any two functions $f, g \in A \to \mathbb{R}$, and number $r \in \mathbb{R}$, functions $f \pm g \in A \to \mathbb{R}$ given by $a \mapsto f(a) \pm g(a)$, maps a to addition/subtraction of $f(a)$ and $g(a)$, are pointwise addition/subtraction of f and g, function $rf \in A \to \mathbb{R}$ given by $a \mapsto rf(a)$ is the scalar product of r and f, and function $f + r \in A \to \mathbb{R}$ given by $a \mapsto f(a) + r$ is f shifted by r. For any set of variables X, we denote the set of *functions* and *continuous functions* from $\mathbb{R}^X$ to $\mathbb{R}$ by $\mathbb{F}_X$ and $\mathbb{C}_X$, respectively.

Let X and Y be two arbitrary disjoint sets of variables. For any two points $\nu_1 \in \mathbb{R}^X$ and $\nu_2 \in \mathbb{R}^Y$, concatenation of ν_1 with ν_2, denoted by $\nu_1 \frown \nu_2$, is defined to be $\nu \in \mathbb{R}^{X \cup Y}$ that maps x to $\nu_1(x)$ if $x \in X$ and to $\nu_2(x)$, otherwise (note that $\nu_1 \frown \nu_2 = \nu_2 \frown \nu_1$). Similarly,

for any two sets of points $\mathtt{V}_1 \subseteq \mathbb{R}^X$ and $\mathtt{V}_2 \subseteq \mathbb{R}^Y$, $\mathtt{V}_1 \frown \mathtt{V}_2 \coloneqq \{\nu_1 \frown \nu_2 \mid \nu_1 \in \mathtt{V}_1, \nu_2 \in \mathtt{V}_2\}$. We use X' to denote the *primed* version of X (*i.e.* for every $x \in X$, variable x' belongs to X').

For any two numbers $a, b \in \mathbb{R}$, we define $[a, b] \coloneqq \{x \in \mathbb{R} \mid a \leq x \leq b\}$ to be the *interval* of points between a and b. We use $\mathcal{I}$ to denote the set of intervals.

2.2 Extended Metric Space and Distance Functions

Let M be an arbitrary set and $d \in M \times M \to \mathbb{R} \cup \{\infty\}$ be an arbitrary function. Ordered pair (M, d) is called an *extended metric space* and d is called a *distance function* iff for any $x, y, z \in M$ the following conditions hold:

1. $d(x, y) \geq 0$,
2. $d(x, y) = 0 \Leftrightarrow x = y$,
3. $d(x, y) = d(y, x)$, and
4. $d(x, z) \leq d(x, y) + d(y, z)$.

Let X be a finite set of variables, and $M \subseteq \mathbb{R}^X$ be an arbitrary set. A well-known distance function on M, denoted by $\mathtt{d}_\infty(\nu_1, \nu_2)$, maps any two points $\nu_1, \nu_2 \in M$ to $\max\limits_{x \in X} |\nu_1(x) - \nu_2(x)|$.

Let $f \in M \times M \to \mathbb{R}_{\geq 0} \cup \{\infty\}$ be an arbitrary function. For any point $p \in M$ and set $A \subseteq M$, we define $f^{\to}(p, A)$ to be $\inf\limits_{a \in A} f(p, a)$; note that if $A = \emptyset$, this means that $f^{\to}(p, A) = \infty$. Intuitively, if f is a distance function then $f^{\to}(p, A)$ is the distance of p from A. Furthermore, for any set $B \subseteq M$, $f^{\to}(A, B)$ is defined to be $\sup\limits_{a \in A} f^{\to}(a, B)$. This means that when $A = \emptyset$, $f^{\to}(A, B) = 0$. Intuitively, if f is a distance function then $f^{\to}(A, B)$ is the *asymmetric distance* from A to B [1]. For any $\epsilon \in \mathbb{R}_{\geq 0}$, $\mathtt{B}^{\epsilon}_{\infty}(A) \coloneqq \{p \in M \mid \mathtt{d}^{\to}_{\infty}(p, A) \leq \epsilon\}$ is the ϵ-ball around A. Also, *closure* of A is denoted by $\mathtt{cl}(A)$ and is defined to be $\mathtt{B}^{0}_{\infty}(A)$. A set is *closed* iff it is equal to its closure. We say $A \subseteq M$ is *bounded* iff $\sup\limits_{a_1, a_2 \in A} \mathtt{d}_\infty(a_1, a_2) \in \mathbb{R}$. Finally, a set is *compact* iff it is closed and bounded.

2.3 Predicates and Perturbations

In this paper we will consider predicates described in first order logic, where the assignment to free variables is constrained to be within a bounded, closed interval. We fix $\mathbb{I}$ to be this interval bounding the domain of all variables, for the rest of this paper. Further for a finite set of variables X, $\mathbb{B}_X$ will denote $\mathbb{I}^X$, *i.e.* the box of dimension $|X|$ defined by interval $\mathbb{I}$.

Let us fix a finite set of variables X. An *atomic predicate with free variables in* X is a constraint of the form $f \geq 0$, where $f \in \mathbb{C}_X$ is a continuous function. We denote the set of atomic propositions by $\mathbb{P}$ and those with X as their set of free variables by $\mathbb{P}_X$. For any two functions $f, g \in \mathbb{F}_X$, we use $f \geq g$, $f \leq g$, and $f = g$, to denote $f - g \geq 0$, $g - f \geq 0$, and $\min(f - g, g - f) \geq 0$, respectively. A *predicate* is defined according to the following BNF rules, where $f \in \mathbb{C}$ is an arbitrary continuous function, x is an arbitrary variable, and $\mathtt{I} \subseteq \mathbb{I}$ is an arbitrary interval:

$$\varphi ::= f \geq 0 \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \exists x \in \mathtt{I} \bullet \varphi \mid \forall x \in \mathtt{I} \bullet \varphi \mid \varphi(X'/X)$$

To simplify notation, instead of writing $\exists x_1, x_2, \ldots, x_n \in \mathtt{I} \bullet \varphi$ and $\forall x_1, x_2, \ldots, x_n \in \mathtt{I} \bullet \varphi$, we write $\exists Y \in \mathtt{I} \bullet \varphi$ and $\forall Y \in \mathtt{I} \bullet \varphi$, where $Y \coloneqq \{x_1, x_2, \ldots, x_n\}$. Formula $\varphi(X'/X)$ is the usual substitution of every free variable $x \in X$ by x', and can be found in a standard logic book like [5, pp. 45-51]. We implicitly assume that before carrying out the substitution, all

[1] $\max\{f^{\to}(A, B), f^{\to}(B, A)\}$ is the *Hausdorff distance* between A and B [19, Section 7.3].

bound variables in φ are renamed using fresh variables that are also distinct from X'. For any predicate φ, the set of *free variables* of φ, denoted by $\texttt{fvar}(\varphi)$, is inductively defined according to the following rules:

1. $\texttt{fvar}(f)$ is X, where f is a function from $\mathbb{R}^X$ to $\mathbb{R}$,
2. $\texttt{fvar}(\varphi \vee \psi)$ and $\texttt{fvar}(\varphi \wedge \psi)$ are defined to be $\texttt{fvar}(\varphi) \cup \texttt{fvar}(\psi)$,
3. $\texttt{fvar}(\exists y \in \texttt{I} \bullet \varphi)$ and $\texttt{fvar}(\forall y \in \texttt{I} \bullet \varphi)$ are defined to be $\texttt{fvar}(\varphi) \setminus \{y\}$, and
4. $\texttt{fvar}(\varphi(X'/X))$ is defined to be $(\texttt{fvar}(\varphi) \setminus X) \cup (\texttt{fvar}(\varphi) \cap X)'$.

For any predicate φ with $X \coloneqq \texttt{fvar}(\varphi)$, we use $[\![\varphi]\!]$ to refer to the subset of points in the box $\mathbb{B}_X$ that satisfy the predicate φ. We denote the set of predicates with Φ and those with X as their set of free variables by Φ_X. We may write $\varphi(X)$ to emphasize $\texttt{fvar}(\varphi) = X$. Also, for any point $\nu \in \mathbb{R}^X$, we use $\nu(X'/X)$ to denote the same point in $\mathbb{R}^{X'}$, *i.e.* a function that maps x' to $\nu(x)$.

For any predicate $\varphi \in \Phi$ and value $\delta \in \mathbb{R}$, *perturbation of φ by δ* is denoted by φ^δ and is a predicate constructed from φ by replacing all atomic predicates of the form $f \geq 0$ with $f + \delta \geq 0$. Note that $[\![\varphi]\!] = [\![\varphi^0]\!]$, and for any $\delta_1 \leq \delta_2 \in \mathbb{R}$, we have $[\![\varphi^{\delta_1}]\!] \subseteq [\![\varphi^{\delta_2}]\!]$. For any two predicates $\varphi_1 \in \Phi_X$ and $\varphi_2 \in \Phi_Y$, and $\delta \in \mathbb{R}$, it is easy to see that $[\![(\varphi_1 \wedge \varphi_2)^\delta]\!] = [\![\varphi_1^\delta \wedge \varphi_2^\delta]\!]$ and $[\![(\varphi_1 \vee \varphi_2)^\delta]\!] = [\![\varphi_1^\delta \vee \varphi_2^\delta]\!]$. Furthermore, if $X = Y$ then $[\![\varphi_1^\delta \wedge \varphi_2^\delta]\!] = [\![\varphi_1^\delta]\!] \cap [\![\varphi_2^\delta]\!]$ and $[\![\varphi_1^\delta \vee \varphi_2^\delta]\!] = [\![\varphi_1^\delta]\!] \cup [\![\varphi_2^\delta]\!]$.

▶ **Definition 1** (Arbitrary Over-Approximation)**.** For any predicate $\varphi \in \Phi$, we say *φ can be arbitrarily over-approximated* iff the following is true:

$$\forall \epsilon \in \mathbb{R}_+ \bullet \exists \delta \in \mathbb{R}_+ \bullet [\![\varphi^\delta]\!] \subseteq \texttt{B}^\epsilon_\infty([\![\varphi]\!])$$

We end this section with an important result that is used many times in our proofs.

▶ **Theorem 2** (Bolzano-Weierstrass [2])**.** *For any finite set of variables X, a bounded set $M \subset \mathbb{R}^X$, and τ an infinite sequence of points in M, τ has a convergent subsequence.*

2.4 Transition Systems and Hybrid Automata

▶ **Definition 3** (Transition Systems)**.** A transition system $\mathcal{T}$ is a tuple $(\texttt{S}, \Sigma, \rightarrow, \texttt{S}^{\mathsf{init}})$ in which

- $\texttt{S}$ is a (possibly infinite) set of *states*,
- Σ is a (possibly infinite) set of *labels*,
- $\rightarrow \subseteq \texttt{S} \times \Sigma \times \texttt{S}$ is a *transition relation*, and
- $\texttt{S}^{\mathsf{init}} \subseteq \texttt{S}$ is a set of *initial states*.

We denote different elements of $\mathcal{T}$ by adding a subscript to their names. For example, we use $\texttt{S}_\mathcal{T}$ to denote the set of states of $\mathcal{T}$. We may omit the subscript whenever it is clear from the context. We write $s \xrightarrow{\sigma} s'$ instead of $(s, \sigma, s') \in \rightarrow$, and $s \rightarrow s'$ to denote $\exists \sigma \in \Sigma \bullet s \xrightarrow{\sigma} s'$.

▶ **Definition 4** (Syntax of Hybrid Automata)**.** A hybrid automaton $\mathcal{H}$ is specified by a tuple $(\texttt{Q}, \texttt{X}, \texttt{E}, \texttt{T}, \texttt{I}, \texttt{F}, \texttt{S}, \texttt{D}, \texttt{R}, \texttt{Q}^{\mathsf{init}}, \texttt{X}^{\mathsf{init}})$ in which

- $\texttt{Q}$ is a non-empty finite set of *locations*.
- $\texttt{X}$ is a non-empty finite set of *variables* that does not contain variable t.
- $\texttt{E}$ is a finite set of *edges*.
- $\texttt{T} \in \mathbb{R}_{\geq 0}$ is a *continuous transition time-bound*. *Wlog.*, we assume $[0, \texttt{T}] \subseteq \mathbb{I}$.
- $\texttt{I} \in \texttt{Q} \rightarrow \Phi_\texttt{X}$ maps each location to a predicate as the *invariant* of that location.
- $\texttt{F} \in \texttt{Q} \rightarrow \Phi_{\texttt{X} \cup \texttt{X}' \cup \{t\}}$, maps each location to a predicate as the *flow* of that location.
- $\texttt{S}, \texttt{D} \in \texttt{E} \rightarrow \texttt{Q}$ map each edge to its *source* and *destination* locations, respectively.
- $\texttt{R} \in \texttt{E} \rightarrow \Phi_{\texttt{X} \cup \texttt{X}'}$ maps each edge to its *transition relation*.

- $\mathtt{Q}^{\mathsf{init}} \in 2^{\mathtt{Q}}$ is a set of *initial locations*.
- $\mathtt{X}^{\mathsf{init}} \in \mathtt{Q}^{\mathsf{init}} \to \Phi_{\mathtt{X}}$ maps each initial location to a predicate as the *initial valuations* of that location.

We denote different elements of $\mathcal{H}$ by adding a subscript to their names. For example, we use $\mathtt{X}_{\mathcal{H}}$ to denote the set of variables of $\mathcal{H}$. We may omit the subscript whenever it is clear from the context. Finally, we define $\mathtt{Inv}(\mathcal{H}) \coloneqq \bigvee_{q\in\mathtt{Q}} \mathtt{I}(q)$ to be the union of invariants in $\mathcal{H}$. Note that $\mathtt{Inv}(\mathcal{H}) \in \Phi_{\mathtt{X}}$.

▶ **Definition 5** (Semantics of Hybrid Automata)**.** Semantics of a hybrid automaton $\mathcal{H}$ is defined using the transition system $[\![\mathcal{H}]\!] = (\mathtt{S}, \Sigma, \to, \mathtt{S}^{\mathsf{init}})$ in which

- $\mathtt{S} \coloneqq \mathtt{Q} \times \mathbb{B}_{\mathtt{X}}$,
- $\Sigma \coloneqq \mathtt{E} \cup [0, \mathtt{T}]$ [2],
- $\mathtt{S}^{\mathsf{init}} \coloneqq \{(q,\nu) \mid q \in \mathtt{Q}^{\mathsf{init}} \wedge \nu \in [\![\mathtt{X}^{\mathsf{init}}(q) \wedge \mathtt{I}(q)]\!]\}$, and
- $\to \coloneqq \to_c \cup \to_d$ where
  - $\to_c$ is the set of *continuous* transitions and for any $r \in \mathbb{R}$ we have $(q,\nu) \xrightarrow{r}_c (q',\nu')$ iff
    1. $r \in [0, \mathtt{T}]$,
    2. $q = q'$,
    3. $\nu \in [\![\mathtt{I}(q)]\!]$,
    4. $\nu' \in [\![\mathtt{I}(q')]\!]$, and
    5. $\nu \frown \nu'(\mathtt{X}'/\mathtt{X}) \frown \{t \mapsto r\} \in [\![\mathtt{F}(q)]\!]$,
  - $\to_d$ is the set of *discrete* transitions and for any $e \in \mathtt{E}$ we have $(q,\nu) \xrightarrow{e}_d (q',\nu')$ iff
    1. $q = \mathtt{S}(e)$,
    2. $q' = \mathtt{D}(e)$,
    3. $\nu \in [\![\mathtt{I}(q)]\!]$,
    4. $\nu' \in [\![\mathtt{I}(q')]\!]$, and
    5. $\nu \frown \nu'(\mathtt{X}'/\mathtt{X}) \in [\![\mathtt{R}(e)]\!]$.

For any two states $s_1 \coloneqq (q_1,\nu_1), s_2 \coloneqq (q_2,\nu_2) \in \mathtt{S}$, we extend definition of $\mathtt{d}_\infty(s_1,s_2)$ to be ∞ if $q_1 \neq q_2$ and $\mathtt{d}_\infty(\nu_1,\nu_2)$ otherwise. For any state $s \in \mathtt{S}$, we define $\mathtt{CPost}_{\mathcal{H}}(s) \coloneqq \{s' \in \mathtt{S} \mid \exists t \in [0,\mathtt{T}] \bullet s \xrightarrow{t} s'\}$. Similarly, for any state $s \in \mathtt{S}$ and $e \in \mathtt{E}$, we define $\mathtt{DPost}^e_{\mathcal{H}}(s) \coloneqq \{s' \in \mathtt{S} \mid s \xrightarrow{e} s'\}$. Also, for any set of states $S \subseteq \mathtt{S}$, we define $\mathtt{CPost}_{\mathcal{H}}(S) \coloneqq \bigcup_{s\in S} \mathtt{CPost}_{\mathcal{H}}(s)$, and $\mathtt{DPost}^e_{\mathcal{H}}(S) \coloneqq \bigcup_{s\in S} \mathtt{DPost}^e_{\mathcal{H}}(s)$. To simplify notation, we may drop the subscript $\mathcal{H}$ if it is clear from the context. Finally, for any number $k \in \mathbb{N}$, we define $\mathtt{reach}_k(\mathcal{H})$ to be the set of states in $\mathtt{S}$ that can be reached from $\mathtt{S}^{\mathsf{init}}$ through *at most* k function applications of $\mathtt{CPost}$ or $\mathtt{DPost}$.

Next, we define a notion of distance between hybrid automata and a related notion of simulation equivalence.

▶ **Definition 6** (k-Step Asymmetric Distance)**.** For any two hybrid automata $\mathcal{H}_1$ and $\mathcal{H}_2$ with the same set of locations, variables, and edges, and for any two states $s_1 \in \mathtt{S}$, and $s_2 \in \mathtt{S}$, we define $\mathtt{adist}_0(s_1,s_2)$ to be $\mathtt{d}_\infty(s_1,s_2)$. For any $k \in \mathbb{N}_+$, we inductively define $\mathtt{adist}_k(s_1,s_2)$ to be maximum of the following values:

$$\begin{aligned} &\mathtt{d}_\infty(s_1,s_2) \\ &\mathtt{adist}^{\to}_{k-1}(\mathtt{CPost}_{\mathcal{H}_1}(s_1), \mathtt{CPost}_{\mathcal{H}_2}(s_2)) \\ &\max_{e\in\mathtt{E}} \mathtt{adist}^{\to}_{k-1}\big(\mathtt{DPost}^e_{\mathcal{H}_1}(s_1), \mathtt{DPost}^e_{\mathcal{H}_2}(s_2)\big) \end{aligned}$$

[2] *Wlog.* we assume $\mathtt{E}$ and $\mathbb{R}$ are disjoint.

Whenever, $\mathcal{H}_1$ and/or $\mathcal{H}_2$ are not clear from the context, we use $\underset{\mathcal{H}_1,\mathcal{H}_2}{\mathtt{adist}}{}^{\rightarrow}_{k}$. Finally, we define $\mathtt{adist}^{\rightarrow}_{k}(\mathcal{H}_1,\mathcal{H}_2)$ to be $\underset{\mathcal{H}_1,\mathcal{H}_2}{\mathtt{adist}}{}^{\rightarrow}_{k}(\mathtt{S}^{\mathsf{init}}_{\mathcal{H}_1},\mathtt{S}^{\mathsf{init}}_{\mathcal{H}_2})$.

▶ **Definition 7** (k-Step ϵ-Simulation)**.** For any two hybrid automata $\mathcal{H}_1$ and $\mathcal{H}_2$ with the same set of locations, variables, and edges, and value $\epsilon \in \mathbb{R}_{\geq 0}$, a relation $R \subseteq \mathtt{S} \times \mathtt{S}$ is called a 0-*step* ϵ-*simulation* iff $\forall s_1, s_2 \in \mathtt{S} \bullet s_1 R s_2 \Rightarrow \mathtt{d}_\infty(s_1, s_2) < \epsilon$. For any $k \in \mathbb{N}_+$, $R \subseteq \mathtt{S} \times \mathtt{S}$ is called a k-*step* ϵ-*simulation* iff there exists $R' \subseteq \mathtt{S} \times \mathtt{S}$, a $k-1$-step ϵ-simulation, such that for any $s_1, s_2 \in \mathtt{S}$, if $s_1 R s_2$ then $\mathtt{d}_\infty(s_1, s_2) < \epsilon$ and both the following conditions hold:

- $\forall s_1' \in \mathtt{S}, e \in \mathtt{E} \bullet s_1 \xrightarrow{e} s_1' \Rightarrow \exists s_2' \in \mathtt{S} \bullet s_2 \xrightarrow{e} s_2' \wedge s_1' R' s_2'$
- $\forall s_1' \in \mathtt{S}, t \in \mathbb{R} \bullet s_1 \xrightarrow{t} s_1' \Rightarrow \exists s_2' \in \mathtt{S}, t' \in \mathbb{R} \bullet s_2 \xrightarrow{t'} s_2' \wedge s_1' R' s_2'$

We say $\mathcal{H}_1$ is k-*step* ϵ-*similar* to $\mathcal{H}_2$, denoted by $\mathcal{H}_1 \preceq^{\epsilon}_{k} \mathcal{H}_2$ iff there is a k-step ϵ-simulation relation R such that $\forall s_1 \in \mathtt{S}^{\mathsf{init}}_{\mathcal{H}_1} \bullet \exists s_2 \in \mathtt{S}^{\mathsf{init}}_{\mathcal{H}_2} \bullet s_1 R s_2$.

▶ **Proposition 8** (Equivalence of k-Step Distance and ϵ-Simulation)**.** *For any two hybrid automata $\mathcal{H}_1$ and $\mathcal{H}_2$, numbers $k \in \mathbb{N}$ and $\epsilon \in \mathbb{R}_+$, $\mathcal{H}_1 \preceq^{\epsilon}_{k} \mathcal{H}_2$ iff $\mathtt{adist}^{\rightarrow}_{k}(\mathcal{H}_1, \mathcal{H}_2) < \epsilon$.*

▶ **Definition 9** (Perturbation of hybrid automata)**.** For any hybrid automaton $\mathcal{H}$ and perturbation $\delta \in \mathbb{R}_{\geq 0}$, perturbation of $\mathcal{H}$ by δ, denoted by $\mathcal{H}^\delta$, is obtained from $\mathcal{H}$ by perturbing all of its predicates by δ. More precisely, $\mathtt{Q}_{\mathcal{H}^\delta} \coloneqq \mathtt{Q}_{\mathcal{H}}$, $\mathtt{X}_{\mathcal{H}^\delta} \coloneqq \mathtt{X}_{\mathcal{H}}$, $\mathtt{E}_{\mathcal{H}^\delta} \coloneqq \mathtt{E}_{\mathcal{H}}$, $\mathtt{T}_{\mathcal{H}^\delta} \coloneqq \mathtt{T}_{\mathcal{H}}$, $\mathtt{S}_{\mathcal{H}^\delta} \coloneqq \mathtt{S}_{\mathcal{H}}$, $\mathtt{D}_{\mathcal{H}^\delta} \coloneqq \mathtt{D}_{\mathcal{H}}$, and $\mathtt{Q}^{\mathsf{init}}_{\mathcal{H}^\delta} \coloneqq \mathtt{Q}^{\mathsf{init}}_{\mathcal{H}}$. Furthermore, for any $q \in \mathtt{Q}$ and $e \in \mathtt{E}$, we have $\mathtt{I}_{\mathcal{H}^\delta}(q) \coloneqq (\mathtt{I}_{\mathcal{H}}(q))^\delta$, $\mathtt{F}_{\mathcal{H}^\delta}(q) \coloneqq (\mathtt{F}_{\mathcal{H}}(q))^\delta$, $\mathtt{R}_{\mathcal{H}^\delta}(e) \coloneqq (\mathtt{R}_{\mathcal{H}}(e))^\delta$, and if $q \in \mathtt{Q}^{\mathsf{init}}$ then $\mathtt{X}^{\mathsf{init}}_{\mathcal{H}^\delta}(q) \coloneqq (\mathtt{X}^{\mathsf{init}}_{\mathcal{H}}(q))^\delta$.

It is easy to see, for any hybrid automaton $\mathcal{H}$, perturbation $\delta \in \mathbb{R}_{\geq 0}$, states $s, s' \in \mathtt{S}$, time $t \in \mathbb{R}$, and edge $e \in \mathtt{E}$, we have $s \xrightarrow{t}_{[\![\mathcal{H}]\!]} s'$ implies $s \xrightarrow{t}_{[\![\mathcal{H}^\delta]\!]} s'$, and $s \xrightarrow{e}_{[\![\mathcal{H}]\!]} s'$ implies $s \xrightarrow{e}_{[\![\mathcal{H}^\delta]\!]} s'$. Also, if $\delta = 0$ then $[\![\mathcal{H}^\delta]\!] = [\![\mathcal{H}]\!]$. Finally, for any $k \in \mathbb{N}$, it is easy to see $\mathtt{adist}^{\rightarrow}_{k}(\mathcal{H}, \mathcal{H}^\delta) = 0$. However, $\mathtt{adist}^{\rightarrow}_{k}(\mathcal{H}^\delta, \mathcal{H})$ is not necessarily 0.

2.5 Encoding States, CPost, and DPost as Predicates

Since in Section 2.3, we only allow variables to be quantified over intervals, *wlog.*, for the rest of this paper and for any hybrid automaton $\mathcal{H}$, we assume

1. $\mathtt{Q}_{\mathcal{H}}$ is the set $\{0, 1, \ldots, |\mathtt{Q}| - 1\}$, and
2. $\mathtt{X}_{\mathcal{H}}$ contains variable $x_{\mathtt{Q}}$.

Variable $x_{\mathtt{Q}}$ is used to model the current location. For any location $q \in \mathtt{Q}$ and edge $e \in \mathtt{E}$, *wlog.*, we assume

- $[\![\mathtt{I}(q)]\!] = [\![x_{\mathtt{Q}} = q \wedge \mathtt{I}(q)]\!]$
- $[\![\mathtt{F}(q)]\!] = [\![x_{\mathtt{Q}} = q \wedge \mathtt{F}(q)]\!]$
- $[\![\mathtt{R}(e)]\!] = [\![x_{\mathtt{Q}} = \mathtt{S}(e) \wedge x'_{\mathtt{Q}} = \mathtt{D}(e) \wedge \mathtt{R}(e)]\!]$
- $q \in \mathtt{Q}^{\mathsf{init}} \Rightarrow [\![\mathtt{X}^{\mathsf{init}}(q)]\!] = [\![x_{\mathtt{Q}} = q \wedge \mathtt{X}^{\mathsf{init}}(q)]\!]$

Let $\varphi \in \Phi_{\mathtt{X}}$ be a predicate encoding a subset of states in $\mathtt{S}_{[\![\mathcal{H}]\!]}$. For any edge $e \in \mathtt{E}$, we define $\mathtt{DPost}^{e}_{\mathcal{H}}(\varphi)$ to be $(\exists \mathtt{X} \in \mathbb{I} \bullet \varphi \wedge \psi_1(\mathtt{X}) \wedge \psi_2(\mathtt{X}') \wedge \psi_3(\mathtt{X} \cup \mathtt{X}'))(\mathtt{X}/\mathtt{X}')$, where

1. $\psi_1 \coloneqq \mathtt{I}(\mathtt{S}(e))$ ensures source location is correct and its invariant is satisfied,
2. $\psi_2 \coloneqq \mathtt{I}(\mathtt{D}(e))(\mathtt{X}'/\mathtt{X})$ ensures destination location is correct and its invariant is satisfied, and
3. $\psi_3 \coloneqq \mathtt{R}(e)$ ensures transition relation is satisfied.

At the end of formula, substituting every free variable $x' \in \mathtt{X}'$ by $x \in \mathtt{X}$, ensures all free variables of $\mathtt{DPost}^e_{\mathcal{H}}(\varphi)$ belong to $\mathtt{X}$ [3]. Note that $\mathtt{DPost}^e_{\mathcal{H}}(\varphi)$ belongs to $\Phi_{\mathtt{X}}$.

For any $q \in \mathtt{Q}$, we define $\mathtt{CPost}^q_{\mathcal{H}}(\varphi)$ to be $(\exists \mathtt{X} \in \mathbb{I}, t \in [0, \mathtt{T}] \bullet \varphi \wedge \psi_1(\mathtt{X}) \wedge \psi_2(\mathtt{X}') \wedge \psi_3(\mathtt{X} \cup \mathtt{X}' \cup \{t\}))(\mathtt{X}/\mathtt{X}')$, where

1. $\psi_1 \coloneqq \mathtt{I}(q)$ ensures source location is correct and its invariant is satisfied,
2. $\psi_2 \coloneqq \mathtt{I}(q)(\mathtt{X}'/\mathtt{X})$ ensures location does not change and its invariant will remain satisfied, and
3. $\psi_3 \coloneqq \mathtt{F}(q)$ ensures flow relation is satisfied.

At the end of formula, substituting every free variable $x' \in \mathtt{X}'$ by $x \in \mathtt{X}$, ensures all free variables of $\mathtt{CPost}^q_{\mathcal{H}}(\varphi)$ belong to $\mathtt{X}$. We also define $\mathtt{CPost}_{\mathcal{H}}(\varphi)$ to be $\bigvee_{q \in \mathtt{Q}} \mathtt{CPost}^q_{\mathcal{H}}(\varphi)$. Note that $\mathtt{CPost}^q_{\mathcal{H}}(\varphi)$ and $\mathtt{CPost}_{\mathcal{H}}(\varphi)$ are both members of $\Phi_{\mathtt{X}}$. Finally, it is easy to see that for any $e \in \mathtt{E}$ and $\varphi \in \Phi_{\mathtt{X}}$ we have $[\![\mathtt{DPost}^e_{\mathcal{H}}(\varphi)]\!] = \mathtt{DPost}^e_{\mathcal{H}}([\![\varphi]\!])$ and $[\![\mathtt{CPost}_{\mathcal{H}}(\varphi)]\!] = \mathtt{CPost}_{\mathcal{H}}([\![\varphi]\!])$ [9, 12].

3 Arbitrary Over-Approximation of a Predicate

The crux of our result on the relation between syntactic and semantic perturbations of hybrid automata relies on the observation that we can arbitrarily over-approximate any predicate, which is formalized in Theorem 10.

▶ **Theorem 10.** *For any set of variables X and predicate $\varphi \in \Phi_X$, φ can be arbitrarily over-approximated.*

The proof is by induction on the structure of φ, and relies on an important topological property of the formulas $\varphi \in \Phi_X$, namely, that the set of satisfiable assignments represented by the formulas is closed. The proofs have been moved to the appendix in the interest of space.

▶ **Theorem 11.** *For any finite set of variables X and predicate $\varphi \in \Phi_X$, the set $[\![\varphi]\!]$ is closed.*

Note that requiring quantified variables to range over bounded intervals is necessary here. For example, let $P \coloneqq (xy = 1)$. Clearly $[\![P]\!]$ is a closed set. However, $[\![\exists y \in \mathbb{R} \bullet P]\!] = \{x \in \mathbb{I} \mid x \neq 0\}$ may not be closed anymore (if $0 \in \mathbb{I}$). Also, even though we do not allow strict inequalities in the syntax, there are typically ways in which a constraint like $f(X) > 0$ can be encoded. However, these different ways are effectively ruled out. Here are some examples to illustrate this point.

1. The formula $\exists \epsilon \in \mathbb{R}_+ \bullet f(X) - \epsilon \geq 0$ is ruled out because we only allow quantification over closed intervals.
2. Consider $\exists y \in [0, 1] \bullet 0 \leq y \leq 1 \wedge f(X) - g(y) \geq 0$, where $g(y) = 1$ if $y = 0$, and $g(y) = y$ if $y \neq 0$. This is also not allowed because the function g is discontinuous.
3. Finally, $\exists y \in \mathbb{R} \bullet y \geq 1 \wedge f(X) - \frac{1}{y} \geq 0$ is ruled out because we only allow quantification over bounded intervals.

[3] As we mentioned in Section 2.3, in order to prevent variable capture that may happen as a result of a substitution, before every substitution, all bound variables are renamed to some fresh variable distinct from those in $\mathtt{X}$ or $\mathtt{X}'$.

4 Arbitrary Over-Approximation of Hybrid Automata

In Section 2.4, we defined hybrid automata, its semantics, its perturbation, as well as bounded step distance function. In Theorem 10, we proved that for any set of variables X, every predicate in Φ_X can be arbitrarily over-approximated. Our main result of this section is that for any bounded number of steps, any hybrid automaton can be arbitrarily over-approximated. In other words, it is always possible to make sure a hybrid automaton and its perturbation behave similarly for at least k steps. Theorem 14 formally states this result. Before, we present this theorem and its proof, we will introduce two lemmas that will help us prove our main result.

▶ **Lemma 12** (CPost and DPost are Continuous). *For any hybrid automaton $\mathcal{H}$ and edge $e \in \mathtt{E}$, functions $\mathtt{CPost}_{\mathcal{H}}$ and $\mathtt{DPost}^e_{\mathcal{H}}$ are continuous with respect to $\mathtt{d}^{\rightarrow}_{\infty}$ and perturbation. More precisely,*

$$\forall \nu \in [\![\mathtt{Inv}(\mathcal{H})]\!], \epsilon \in \mathbb{R}_+ \bullet \exists \delta \in \mathbb{R}_+ \bullet \forall \nu' \in [\![\mathtt{Inv}(\mathcal{H}^\delta)]\!] \bullet \mathtt{d}_\infty(\nu', \nu) \leq \delta \Rightarrow$$
$$\mathtt{d}^{\rightarrow}_{\infty}(\mathtt{CPost}_{\mathcal{H}^\delta}(\nu'), \mathtt{CPost}_{\mathcal{H}}(\nu)) < \epsilon \wedge \mathtt{d}^{\rightarrow}_{\infty}(\mathtt{DPost}^e_{\mathcal{H}^\delta}(\nu'), \mathtt{DPost}^e_{\mathcal{H}}(\nu)) < \epsilon$$

Proof. We prove this lemma for CPost. Proof of $\mathtt{DPost}^e$ can be obtained by replacing every CPost with $\mathtt{DPost}^e$. For the purpose of contradiction suppose this result does not hold for some $\nu \in [\![\mathtt{Inv}(\mathcal{H})]\!]$, $\epsilon \in \mathbb{R}_+$. For any $n \in \mathbb{N}$, define $\delta_n := \frac{1}{n+1}$, and let $\nu'_n \in [\![\mathtt{Inv}(\mathcal{H}^{\delta_n})]\!]$ be a point for which $\mathtt{d}_\infty(\nu'_n, \nu) \leq \delta_n$ and $\mathtt{d}^{\rightarrow}_{\infty}(\mathtt{CPost}_{\mathcal{H}^{\delta_n}}(\nu'_n), \mathtt{CPost}_{\mathcal{H}}(\nu)) \geq \epsilon$ are both true.

Define $\varphi := \bigwedge_{x \in \mathtt{X}}(x = \nu(x))$. Since $\varphi \in \mathbb{P}_{\mathtt{X}}$, we know $\mathtt{CPost}_{\mathcal{H}}(\varphi) \in \Phi_{\mathtt{X}}$. Use Theorem 10 and fix $n \in \mathbb{N}$ such that $[\![(\mathtt{CPost}_{\mathcal{H}}(\varphi))^{\delta_n}]\!] \subseteq \mathtt{B}^{\epsilon/2}_\infty([\![\mathtt{CPost}_{\mathcal{H}}(\varphi)]\!])$. Definition of $\mathtt{CPost}_{\mathcal{H}}(\varphi)$ ensures the following is true, because the two sides of the equality are syntactically the same.

$$\forall \delta \in \mathbb{R}_{\geq 0} \bullet [\![\mathtt{CPost}_{\mathcal{H}^\delta}(\varphi^\delta)]\!] = [\![(\mathtt{CPost}_{\mathcal{H}}(\varphi))^\delta]\!]$$

Therefore, knowing $\forall n \in \mathbb{N} \bullet \nu'_n \in [\![\varphi^{\delta_n}]\!]$, we conclude

$$\mathtt{CPost}_{\mathcal{H}^{\delta_n}}(\nu'_n) \subseteq \mathtt{CPost}_{\mathcal{H}^{\delta_n}}([\![\varphi^{\delta_n}]\!]) = [\![\mathtt{CPost}_{\mathcal{H}^{\delta_n}}(\varphi^{\delta_n})]\!] \subseteq$$
$$\mathtt{B}^{\epsilon/2}_\infty([\![\mathtt{CPost}_{\mathcal{H}}(\varphi)]\!]) = \mathtt{B}^{\epsilon/2}_\infty(\mathtt{CPost}_{\mathcal{H}}(\nu))$$

This means $\mathtt{d}^{\rightarrow}_{\infty}(\mathtt{CPost}_{\mathcal{H}^{\delta_n}}(\nu'_n), \mathtt{CPost}_{\mathcal{H}}(\nu)) \leq \frac{\epsilon}{2}$, which is a contradiction. ◀

▶ **Lemma 13** (adist is Continuous). *For any hybrid automaton $\mathcal{H}$ and number $k \in \mathbb{N}$, distance function $\mathtt{adist}^{\rightarrow}_k$ is continuous with respect to $\mathtt{d}^{\rightarrow}_{\infty}$ and perturbation. More precisely,*

$$\forall \varphi \in \Phi_{\mathtt{X}}, \epsilon \in \mathbb{R}_+ \bullet \exists \delta \in \mathbb{R}_+ \bullet \forall \psi \in \Phi_{\mathtt{X}} \bullet$$
$$[\![\varphi]\!] \subseteq [\![\mathtt{Inv}(\mathcal{H})]\!] \wedge [\![\psi]\!] \subseteq [\![\mathtt{Inv}(\mathcal{H}^\delta)]\!] \wedge \mathtt{d}^{\rightarrow}_{\infty}([\![\psi]\!], [\![\varphi]\!]) \leq \delta \Rightarrow \underset{\mathcal{H}^\delta, \mathcal{H}}{\mathtt{adist}}{}^{\rightarrow}_k([\![\psi]\!], [\![\varphi]\!]) < \epsilon$$

Proof. Proof is by induction on k. Base of induction, where $k = 0$, is trivial because $\mathtt{adist}^{\rightarrow}_0$ and $\mathtt{d}^{\rightarrow}_{\infty}$ are identical. For the purpose of contradiction, suppose the inductive step does not hold for some $\epsilon \in \mathbb{R}_+$ and $\varphi \in \Phi_{\mathtt{X}}$, where $[\![\varphi]\!] \subseteq [\![\mathtt{Inv}(\mathcal{H})]\!]$. For any $n \in \mathbb{N}$, define $\delta_n := \frac{\epsilon}{n+3}$, and let $\psi_n \in \Phi_{\mathtt{X}}$ be a predicate for which $[\![\psi_n]\!] \subseteq [\![\mathtt{Inv}(\mathcal{H}^{\delta_n})]\!]$, $\mathtt{d}^{\rightarrow}_{\infty}([\![\psi_n]\!], [\![\varphi]\!]) \leq \delta_n \leq \frac{\epsilon}{3}$, and $\underset{\mathcal{H}^\delta, \mathcal{H}}{\mathtt{adist}}{}^{\rightarrow}_{k+1}([\![\psi_n]\!], [\![\varphi]\!]) \geq \epsilon$ are all true. We immediately know $[\![\varphi]\!] \neq \emptyset \wedge \forall n \in \mathbb{N} \bullet \wedge [\![\psi_n]\!] \neq \emptyset$. Using the properties of supremum and infimum, for any $n \in \mathbb{N}$, let $\nu'_n \in [\![\psi_n]\!]$ be such that

$$\forall \nu \in [\![\varphi]\!] \bullet \mathtt{adist}_{k+1}(\nu'_n, \nu) \geq \frac{\epsilon}{2}$$

Let $\nu_n \in [\![\varphi]\!] \cap \mathtt{B}^{\delta_n}_\infty(\nu'_n)$ be an arbitrary point (the set $[\![\varphi]\!] \cap \mathtt{B}^{\delta_n}_\infty(\nu'_n)$ is never empty). Using definition of $\mathtt{adist}_{k+1}$, we know at least one of the following conditions is true:

$$\mathtt{adist}^{\rightarrow}_k(\mathtt{CPost}_{\mathcal{H}^{\delta_n}}(\nu'_n), \mathtt{CPost}_{\mathcal{H}}(\nu_n)) \geq \frac{\epsilon}{2}$$
$$\mathtt{adist}^{\rightarrow}_k(\mathtt{DPost}^e_{\mathcal{H}^{\delta_n}}(\nu'_n), \mathtt{DPost}^e_{\mathcal{H}}(\nu_n)) \geq \frac{\epsilon}{2}, \text{ for some } e \in \mathtt{E}$$

We assume the first condition is true. Proof of the other case can be obtained by replacing every $\texttt{CPost}$ with $\texttt{DPost}^e$ in the rest of this proof. Define $\varphi'_n \coloneqq \bigwedge_{x\in\mathtt{X}}(x = \nu_n(x))$ and $\psi'_n \coloneqq \bigwedge_{x\in\mathtt{X}}(x = \nu'_n(x))$. We know $\varphi'_n, \psi'_n \in \mathbb{P}_{\mathtt{X}}$, and hence $\varphi''_n \coloneqq \texttt{CPost}_{\mathcal{H}}(\varphi'_n)$ and $\psi''_n \coloneqq \texttt{CPost}_{\mathcal{H}^{\delta_n}}(\psi'_n)$ are both members of $\Phi_{\mathtt{X}}$. For any $n \in \mathbb{N}$, we know $[\![\varphi''_n]\!] \subseteq [\![\texttt{Inv}(\mathcal{H})]\!]$ and $[\![\psi''_n]\!] \subseteq [\![\texttt{Inv}(\mathcal{H}^{\delta_n})]\!]$ are both true. Use induction hypothesis and find $N \in \mathbb{N}$ such that

$$\forall n \in \mathbb{N} \bullet \mathtt{d}^{\rightarrow}_{\infty}([\![\psi''_n]\!], [\![\varphi''_n]\!]) \leq \delta_N \Rightarrow \underset{\mathcal{H}^{\delta_n},\mathcal{H}}{\mathtt{adist}}{}^{\rightarrow}_{k}([\![\psi''_n]\!], [\![\varphi''_n]\!]) < \frac{\epsilon}{2}$$

We conclude the following which is in contradiction with Lemma 12.

$$\forall n \in \mathbb{N} \bullet \nu_n \in [\![\texttt{Inv}(\mathcal{H})]\!] \wedge \nu'_n \in [\![\texttt{Inv}(\mathcal{H}^{\delta_n})]\!] \wedge \mathtt{d}^{\rightarrow}_{\infty}(\nu'_n, \nu_n) \leq \delta_n \wedge$$
$$\mathtt{d}^{\rightarrow}_{\infty}(\texttt{CPost}_{\mathcal{H}^{\delta_n}}(\nu'_n), \texttt{CPost}_{\mathcal{H}}(\nu_n)) > \delta_N$$

◀

▶ **Theorem 14.** *For any hybrid automaton $\mathcal{H}$, bounded step $n \in \mathbb{N}$, and $\epsilon \in \mathbb{R}_+$, there is $\delta \in \mathbb{R}_+$ such that*

$$\forall k \in \{0, \ldots, n\} \bullet \underset{\mathcal{H}^{\delta},\mathcal{H}}{\mathtt{adist}}{}^{\rightarrow}_{k}\left(\mathtt{S}^{\mathsf{init}}_{[\![\mathcal{H}^{\delta}]\!]}, \mathtt{S}^{\mathsf{init}}_{[\![\mathcal{H}]\!]}\right) < \epsilon$$

Proof. Since n is given in advance and $\underset{\mathcal{H}^{\delta},\mathcal{H}}{\mathtt{adist}}{}^{\rightarrow}_{k}(\cdot,\cdot)$ is a non-decreasing function with respect to k, it is enough to prove this result for only $k \coloneqq n$. Let $\varphi \coloneqq \texttt{Inv}(\mathcal{H}) \wedge \bigvee_{q\in\mathbb{Q}^{\mathsf{init}}_{\mathcal{H}}} \mathtt{X}^{\mathsf{init}}(q)$ be a formula in $\Phi_{\mathtt{X}}$ that represents initial states of $\mathcal{H}$. We know $\mathtt{S}^{\mathsf{init}}_{[\![\mathcal{H}]\!]} = [\![\varphi]\!]$ and $\forall \delta \in \mathbb{R} \bullet \mathtt{S}^{\mathsf{init}}_{[\![\mathcal{H}^{\delta}]\!]} = [\![\varphi^{\delta}]\!]$. Using Lemma 13, find $\eta \in \mathbb{R}_+$ such that $\forall \delta \in \mathbb{R} \bullet \mathtt{d}^{\rightarrow}_{\infty}([\![\varphi^{\delta}]\!], [\![\varphi]\!]) < \eta \Rightarrow \underset{\mathcal{H}',\mathcal{H}}{\mathtt{adist}}{}^{\rightarrow}_{k}([\![\varphi^{\delta}]\!], [\![\varphi]\!]) < \epsilon$. Complete the proof by using Theorem 10 and finding $\delta_k \in \mathbb{R}_+$ such that $\mathtt{d}^{\rightarrow}_{\infty}([\![\varphi^{\delta_k}]\!], [\![\varphi]\!]) < \eta$. ◀

▶ **Corollary 15** (Arbitrary Over-Approximation of Reachable Sets)**.** *For any hybrid automaton $\mathcal{H}$, reachable set of $\mathcal{H}$ can be arbitrarily over-approximated. More precisely,*

$$\forall \epsilon \in \mathbb{R}_+, k \in \mathbb{N} \bullet \exists \delta \in \mathbb{R}_+ \bullet \mathtt{reach}_k\left(\mathcal{H}^{\delta}\right) \subseteq \mathtt{B}^{\epsilon}_{\infty}(\mathtt{reach}_k(\mathcal{H}))$$

Proof. Immediate from Theorem 14. ◀

▶ **Corollary 16** (Bounded ϵ-Simulation)**.** *For any hybrid automaton $\mathcal{H}$, $k \in \mathbb{N}$, and $\epsilon \in \mathbb{R}_+$, there is $\delta \in \mathbb{R}_+$ such that $\mathcal{H}$ and $\mathcal{H}^{\delta}$ ϵ-simulate each other for at least k steps.*

$$\forall \epsilon \in \mathbb{R}_+, k \in \mathbb{N} \bullet \exists \delta \in \mathbb{R}_+ \bullet \mathcal{H}^{\delta} \preceq^{\epsilon}_{k} \mathcal{H} \wedge \mathcal{H} \preceq^{\epsilon}_{k} \mathcal{H}^{\delta}$$

Proof. $\mathcal{H} \preceq^{\epsilon}_{k} \mathcal{H}^{\delta}$ is trivially true for any $\epsilon \in \mathbb{R}_{\geq 0}$ and $k \in \mathbb{N}$. ϵ-simulation of $\mathcal{H}^{\delta}$ by $\mathcal{H}$ for at least k steps is immediate from Theorem 14 and Proposition 8. ◀

5 Applications to Safety Model Checking

In Section 3 and Section 4 we proved some results about the ability to arbitrarily over-approximate sets and hybrid automata. In this section, we demonstrate three applications of those results.

5.1 Co-completeness of δ-Complete Decision Procedures

It is well-known that the first order theory of reals with addition, multiplication, and trigonometric functions is undecidable (one can easily encode integers in this logic) [5, Chapter 3]. Authors in [10, 11] came up with an interesting compromise in their decision procedure for *checking satisfiability* of these formulas. For any formula $\varphi \in \Phi$, their so called δ-complete decision procedure returns either $\texttt{unsat}$ or δ-$\texttt{sat}$. If the output is $\texttt{unsat}$, we know $[\![\varphi]\!] = \emptyset$. However, if the output is δ-$\texttt{sat}$, we only know $[\![\varphi^{\delta}]\!] \neq \emptyset$. Parameter $\delta \in \mathbb{R}_+$

is an input to their algorithm and can be made arbitrary small. The algorithm in [10, 11], imposes one additional constraint on formulas: Every function used in atomic propositions of a formula must be type 2 computable [6, 16, 28]. Intuitively, a function is type 2 computable iff an arbitrary approximation of its value can be computed from arbitrary approximations of its inputs. This condition is stronger than continuity requirement that we have in this paper. Nevertheless, it still includes arithmetic, trigonometric, logarithmic, exponential, absolute value, minimum, and maximum functions as well as solutions of many ordinary differential equations.

Observe that δ-complete decision procedures provide approximate answers to the satisfiability question. So even if a formula φ is unsatisfiable, there is no guarantee that the δ-decision procedure will answer `unsat`. The procedure is guaranteed to answer `unsat` only if it is called with a δ such that both φ *and* φ^δ are unsatisfiable. Does such a δ exist for all unsatisfiable φ? Can it be computed? Theorem 10 answers in the affirmative for both these questions – the value of δ can also be computed by repeatedly calling a δ-decision procedure for smaller and smaller δ. Theorem 17 formalizes this result.

▶ **Theorem 17.** *Let $\varphi \in \Phi$ be an arbitrary predicate with $[\![\varphi]\!] = \emptyset$ and every function is type 2 computable. One can compute $\delta \in \mathbb{R}_+$ for which $[\![\varphi^\delta]\!] = \emptyset$.*

We require that the predicates only use non-strict inequalities, whereas no such restriction is imposed on the inputs to the δ-decision procedures [10, 11]. However, predicates with strict inequalities can be easily handled using predicates with non-strict inequalities. Consider φ, a formula with some strict inequalities, and let φ' be the formula obtained by changing each strict inequality in φ to the corresponding non-strict one. Observe that if a δ-complete decision procedure says φ' is `unsat`, we know `unsat` is a valid output for φ as well. More interestingly, output δ-`sat` for φ' means 2δ-`sat` is a valid output for φ.

5.2 Completeness of dReach

δ-complete decision procedures have been used in [12] to develop a tool called `dReach` for bounded time, bounded step safety model checking of non-linear hybrid automaton $\mathcal{H}$. The tool first takes $\mathcal{H}$ and unsafe predicate $U \in \Phi_{\mathtt{X}}$ as input, and then encodes the safety problem into a satisfiability problem of a formula that is supported by the algorithm in [10, 11]. Finally, it checks its satisfiability and outputs the results. Possible outputs are either `unsat`, which means $\mathtt{reach}_k(\mathcal{H}) \cap [\![U]\!] = \emptyset$, and δ-`sat`, which means $\mathtt{reach}_k(\mathcal{H}^\delta) \cap [\![U^\delta]\!] \neq \emptyset$. However, in case the input hybrid automaton is k-step safe, it is not known if a value of δ exists, such that $\mathtt{reach}_k(\mathcal{H}^\delta) \cap [\![U^\delta]\!]$ remains an empty set. Theorem 18 is an immediate consequence of Corollary 15 and Theorem 17.

▶ **Theorem 18.** *Let $\mathcal{H}$ be a hybrid automaton and $U \in \Phi_{\mathtt{X}}$ be an unsafe predicate. If syntax of $\mathcal{H}$ and U is supported by* `dReach` *then for any $k \in \mathbb{N}$, if $\mathcal{H}$ is k-step safe then one can compute $\delta \in \mathbb{R}_+$ for which* `dReach` *answers* `unsat` *(*i.e. *safe).*

Note that the formulas considered here are more general that what is supported in `dReach`. As argued in the previous section, our restriction to non-strict inequalities does not constrain things when using δ-complete decision procedures. Flows specified using ordinary differential equations in `dReach`, define continuous functions and thus are handled by our results. We could also easily change the formula describing the continuous post predicate by requiring $\forall t \in [0, \mathtt{T}] \bullet \mathtt{I}(q)(f(t))$, to ensure that the continuous state satisfies the invariant at all times during a continuous transition.

δ-complete decision procedures can also be used for counter-example validation in a CEGAR-based model checking framework [26]. Theorem 18 can also be used to provide guarantees of progress for such tools – if a counter-example is spurious then such δ-decision procedures are guaranteed to discover it.

6 Conclusion

We investigated whether syntactic perturbations of hybrid automata whose elements are specified in a logic, are semantically close. Such a result does not hold in general as illustrated by the timed automata in Figures 1a and 1b. We identify a fairly general class of systems for which, for every $\epsilon > 0$ and k, there is a $\delta > 0$ such that δ-syntactic perturbations are ϵ-simulation equivalent upto k-discrete steps. These results are important in the context of providing certain theoretical guarantees on δ-decision procedures and bounded verification using the same procedures.

Our results about the semantic closeness of δ-syntactic perturbations only apply when one considers a bounded number of discrete jumps. It seems unlikely that such a result can be extended when there is no a priori bound on the number of discrete steps; this is illustrated by the timed automaton in Figure 1a. One interesting future direction consists of exploring the computability issues with respect to the continuity property of the semantics. More precisely, given an ϵ, can we effectively compute a δ such that δ syntactic perturbation on the hybrid automaton lead to at most ϵ deviations in the semantics.

References

1 E. Asarin and A. Bouajjani. Perturbed turing machines and hybrid systems. In *LICS*, pages 269–278, 2001.

2 R.G. Bartle and D.R. Sherbert. *Introduction to Real Analysis, 4th Edition*. John Wiley & Sons, Incorporated, 2011.

3 P. Bouyer, N. Markey, and P.-A. Reynier. Robust model-checking of linear-time properties in timed automata. In *Proceedings of LATIN*, pages 238–249, 2006.

4 Patricia Bouyer, Nicolas Markey, and Ocan Sankur. Robust model-checking of timed automata via pumping in channel machines. In *Proceedings of FORMATS*, pages 97–112, 2011.

5 Aaron R. Bradley and Zohar Manna. *The Calculus of Computation: Decision Procedures with Applications to Verification*. Springer-Verlag, 2007.

6 Vasco Brattka, Peter Hertling, and Klaus Weihrauch. *A Tutorial on Computable Analysis*, pages 425–491. Springer New York, New York, NY, 2008.

7 Martin de Wulf, Laurent Doyen, Nicolas Markey, and Jean-François Raskin. Robust safety of timed automata. *Formal Methods in System Design*, 33(1):45–84, 2008.

8 Peter Franek, Stefan Ratschan, and Piotr Zgliczynski. Quasi-decidability of a fragment of the first-order theory of real numbers. *Journal of Automated Reasoning*, 57(2):157–185, Aug 2016.

9 Martin Fränzle. *Analysis of Hybrid Systems: An Ounce of Realism Can Save an Infinity of States*, pages 126–139. Springer Berlin Heidelberg, Berlin, Heidelberg, 1999.

10 Sicun Gao, Jeremy Avigad, and Edmund M. Clarke. δ-completee decision procedures for satisfiability over the reals. In *IJCAR*, pages 286–300, Berlin, Heidelberg, 2012. Springer-Verlag.

11 Sicun Gao, Jeremy Avigad, and Edmund M. Clarke. Delta-decidability over the reals. In *LICS*, pages 305–314. IEEE Computer Society, 2012.

12 Sicun Gao, Soonho Kong, Wei Chen, and Edmund M. Clarke. Delta-complete analysis for bounded reachability of hybrid systems. *CoRR*, abs/1404.7171, 2014.

13 Antoine Girard. Approximately bisimilar abstractions of incrementally stable finite or infinite dimensional systems. In *53rd IEEE Conference on Decision and Control, CDC 2014, Los Angeles, CA, USA, December 15-17, 2014*, pages 824–829, 2014.

14 Vineet Gupta, Thomas A. Henzinger, and Radha Jagadeesan. *Robust timed automata*, pages 331–345. Springer Berlin Heidelberg, Berlin, Heidelberg, 1997.

15 Thomas A. Henzinger and Jean-François Raskin. *Robust Undecidability of Timed and Hybrid Systems*, pages 145–159. Springer Berlin Heidelberg, Berlin, Heidelberg, 2000.

16 K.I. Ko. *Complexity theory of real functions*. Progress in theoretical computer science. Birkhäuser, 1991.

17 Ruggero Lanotte and Simone Tini. Taylor approximation for hybrid systems. In Manfred Morari and Lothar Thiele, editors, *Hybrid Systems: Computation and Control*, pages 402–416, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

18 Ruggero Lanotte and Simone Tini. Taylor approximation for hybrid systems. *Information and Computation*, 205(11):1575–1607, 2007.

19 M. O'Searcoid. *Metric Spaces*. Springer Undergraduate Mathematics Series. Springer London, 2006.

20 Giordano Pola, Antoine Girard, and Paulo Tabuada. Approximately bisimilar symbolic models for nonlinear control systems. *Automatica*, 44(10):2508–2516, 2008.

21 Pavithra Prabhakar and Mahesh Viswanathan. A dynamic algorithm for approximate flow computations. In *Proceedings of the 14th ACM International Conference on Hybrid Systems: Computation and Control, HSCC 2011, Chicago, IL, USA, April 12-14, 2011*, pages 133–142, 2011.

22 Pavithra Prabhakar, Vladimeros Vladimerou, Mahesh Viswanathan, and Geir E. Dullerud. Verifying tolerant systems using polynomial approximations. In *Proceedings of the 30th IEEE Real-Time Systems Symposium, RTSS 2009, Washington, DC, USA, 1-4 December 2009*, pages 181–190, 2009.

23 Anuj Puri. Dynamical properties of timed automata. *Discrete Event Dynamic Systems*, 10(1-2):87–113, 2000.

24 Stefan Ratschan. Quantified constraints under perturbation. *Journal of Symbolic Computation*, 33(4):493–505, 2002.

25 Stefan Ratschan. Efficient solving of quantified inequality constraints over the real numbers. *ACM Trans. Comput. Logic*, 7(4):723–748, 2006.

26 Nima Roohi, Pavithra Prabhakar, and Mahesh Viswanathan. *HARE: A Hybrid Abstraction Refinement Engine for Verifying Non-linear Hybrid Automata*, pages 573–588. Springer, 2017.

27 Nima Roohi, Pavithra Prabhakar, and Mahesh Viswanathan. Robust model checking of timed automata under clock drifts. In *HSCC*, pages 153–162. ACM, 2017.

28 K. Weihrauch. *Computable Analysis: An Introduction*. Texts in Theoretical Computer Science. An EATCS Series. Springer Berlin Heidelberg, 2000.

29 M. De Wulf, L. Doyen, N. Markey, and J.-F. Raskin. Robustness and implementability of timed automata. In *Proceedings of FORMATS*, pages 118–133, 2004.

A Proofs

In this section we present all the proofs that we skipped in the main section of the paper.

▶ **Theorem 11**. *For any finite set of variables X and predicate $\varphi \in \Phi_X$, the set $[\![\varphi]\!]$ is closed.*

Proof. Immediate from an induction on the structure of φ, using Lemma 19, Lemma 20, and Lemma 21 (closedness of finite disjunctions and closedness of finite conjunctions of closed sets are trivial facts). ◀

▶ **Theorem 10**. *For any set of variables X and predicate $\varphi \in \Phi_X$, φ can be arbitrarily over-approximated.*

Proof. Immediate from an induction on the structure of φ using Lemma 22, Lemma 23, Lemma 24, Lemma 25, and Lemma 26. ◀

▶ **Lemma 19.** *For any finite set of variables X and predicate $P \in \mathbb{P}_X$, $[\![P]\!]$ is closed.*

Proof. This is a trivial proof once we noticed all functions in P are continuous and all inequalities in P are non-strict. ◀

▶ **Lemma 20.** *For any finite sets of variables X, variable y, interval $\mathtt{I}$, and predicate $\varphi \in \Phi_X$, if $[\![\varphi]\!]$ is closed then $[\![\exists y \in \mathtt{I} \bullet \varphi]\!]$ is also closed.*

Proof. Let ψ be $\exists y \in \mathtt{I} \bullet \varphi$. If $\mathtt{I}$ is an empty set then $[\![\psi]\!] = \emptyset$. Otherwise, if $y \notin X$ then $[\![\psi]\!] = [\![\phi]\!]$. Otherwise, let $\nu_0, \nu_1, \nu_2, \ldots$ be a sequence of points in $[\![\exists y \in \mathtt{I} \bullet \varphi]\!]$ that converges to ν^*. For every $n \in \mathbb{N}$, there is $\nu'_n \in [\![\varphi]\!]$ such that $\nu'_n(y) \in \mathtt{I}$ and $\forall x \in X \backslash \{y\} \bullet \nu'_n(x) = \nu_n(x)$. We know $[\![\varphi]\!]$ is bounded. Therefore, from Theorem 2 and closeness of $\mathtt{I}$, there is a subsequence with indices $m_0 < m_1 < m_2 < \cdots$ such that $\nu'_{m_0}, \nu'_{m_1}, \nu'_{m_2}, \ldots$ converges to ν'^* with $\nu'^*(y) \in \mathtt{I}$. We know $\nu'^* \in [\![\varphi]\!]$, since it is closed, and ν'^* agrees with ν^* on every variable $x \in X \setminus \{y\}$. Therefore, $\nu^* \in [\![\exists y \in \mathtt{I} \bullet \varphi]\!]$. ◀

▶ **Lemma 21.** *For any finite sets of variables X, variable y, interval $\mathtt{I}$, and predicate $\varphi \in \Phi_X$, if $[\![\varphi]\!]$ is closed then $[\![\forall y \in \mathtt{I} \bullet \varphi]\!]$ is also closed.*

Proof. Let ψ be $\forall y \in \mathtt{I} \bullet \varphi$. If $\mathtt{I}$ is an empty set then $[\![\psi]\!] = \mathbb{B}_{X \setminus \{y\}}$. Otherwise, if $y \notin X$ then $[\![\psi]\!] = [\![\phi]\!]$. Otherwise, let $\nu' \in \mathtt{I}^{\{y\}}$ be an arbitrary point, and let $\nu_0, \nu_1, \nu_2, \ldots$ be a sequence of points in $[\![\forall y \in \mathtt{I} \bullet \varphi]\!]$ that converges to ν^*. For any $n \in \mathbb{N}$, we know $\nu_n \frown \nu' \in [\![\varphi]\!]$. We also know $\nu_0 \frown \nu', \nu_1 \frown \nu', \nu_2 \frown \nu', \ldots$ converges to $\nu^* \frown \nu'$ which, by closedness of $[\![\varphi]\!]$, is a point in $[\![\varphi]\!]$. Therefore, $\nu^* \in [\![\forall y \in \mathtt{I} \bullet \varphi]\!]$, since we put no restriction on ν'. ◀

▶ **Lemma 22.** *For any finite set of variables X, any predicate in $\mathbb{P}_X$ can be arbitrarily over-approximated.*

Proof. For the purpose of contradiction, let $P \in \mathbb{P}_X$ be a predicate with the following condition: $\exists \epsilon \in \mathbb{R}_+ \bullet \forall \delta \in \mathbb{R}_+ \bullet \exists \nu \in [\![P^\delta]\!] \bullet \nu \notin \mathtt{B}^\epsilon_\infty([\![P]\!])$. Fix ϵ, and for any $n \in \mathbb{N}$, let $\delta_n := \frac{1}{n+1}$. Let $\nu_n \in [\![P^{\delta_n}]\!] \setminus \mathtt{B}^\epsilon_\infty([\![P]\!])$ be an arbitrary element. Using Theorem 2, there is a sequence $m_0 < m_1 < \cdots$ such that $\nu_{m_0}, \nu_{m_1}, \ldots$ converges to ν^*.

It is enough to show $\nu^* \in [\![P]\!]$, because there are infinitely many indices n such that $\mathtt{d}_\infty(\nu_{m_n}, \nu^*) \leq \epsilon$ and any one of them contradicts the fact $\nu_{m_n} \notin \mathtt{B}^\epsilon_\infty([\![P]\!])$. Suppose $\nu^* \notin [\![P]\!]$. Let the atomic constraint corresponding to P be $f \geq 0$. Since, $\nu^* \notin [\![P]\!]$ we have $f(\nu^*) < 0$, and since, $\nu_n \in [\![P^{\delta_n}]\!]$, we have $\forall n \in \mathbb{N} \bullet f(\nu_{m_n}) + \delta_{m_n} \geq 0$. Since f is a continuous function, $\exists N_1 \in \mathbb{N} \bullet \forall n > N_1 \bullet |f(\nu_{m_n}) - f(\nu^*)| < -\frac{1}{2} f(\nu^*)$. Let $N_2 \in \mathbb{N}$ be such that $\delta_{N_2} < -\frac{1}{2} f(\nu^*)$ and let $N := \max\{N_1, N_2\}$. We know $\forall n > N \bullet f(\nu_{m_n}) + \delta_{m_n} < \frac{1}{2} f(\nu^*) - \frac{1}{2} f(\nu^*) = 0$ which is a contradiction. ◀

▶ **Lemma 23.** *For any finite sets of variables Y, Z and predicates $\varphi \in \Phi_Y$ and $\psi \in \Phi_Z$, if φ and ψ can be arbitrarily over-approximated then $\varphi \vee \psi$ can be arbitrarily approximated as well.*

Proof. For any $\epsilon \in \mathbb{R}_+$, let $\delta \in \mathbb{R}_+$ be such that both $[\![\varphi^\delta]\!] \subseteq \mathtt{B}^\epsilon_\infty([\![\varphi]\!])$ and $[\![\psi^\delta]\!] \subseteq \mathtt{B}^\epsilon_\infty([\![\psi]\!])$ hold. If $[\![\varphi^\delta \vee \psi^\delta]\!] = \emptyset$ then the proof is complete. Otherwise, let $\nu \in [\![\varphi^\delta \vee \psi^\delta]\!]$ be an arbitrary value. At least one of $\nu\restriction_Y \in [\![\varphi^\delta]\!]$ and $\nu\restriction_Z \in [\![\psi^\delta]\!]$ are true. *Wlog.* assume that the first holds. Let $\nu' \in \mathtt{B}^\epsilon_\infty([\![\varphi]\!])$ be any point for which $\mathtt{d}_\infty(\nu', \nu\restriction_Y) \leq \epsilon$, and let $\nu'' \in \mathbb{R}^{Y \cup Z}$ be the point that maps $y \in Y$ to $\nu'(y)$ and $z \in Z \setminus Y$ to $\nu(z)$. We know $\mathtt{d}_\infty(\nu, \nu'') :=$

$\max_{x\in Y\cup Z}|\nu(x)-\nu''(x)|$. The right hand side is equal to the maximum of $\max_{y\in Y}|\nu(y)-\nu''(y)|$ and $\max_{z\in Z\setminus Y}|\nu(z)-\nu''(z)|=0$, and hence equal to $\max_{y\in Y}|\nu(y)-\nu'(y)|=\mathtt{d}_\infty(\nu\restriction_Y,\nu')\leq\epsilon$. What remains is to show $\nu''\in \mathtt{B}^\epsilon_\infty([\![\varphi\vee\psi]\!])$. Let $u'\in[\![\varphi]\!]$ be any point with $\mathtt{d}_\infty(\nu',u')\leq\epsilon$, and let $u''\in\mathbb{R}^{Y\cup Z}$ be the point that maps $y\in Y$ to $u'(x)$ and $z\in Z\setminus Y$ to $\nu''(z)$. Clearly, $\mathtt{d}_\infty(u'',\nu'')=\mathtt{d}_\infty(u',\nu')\leq\epsilon$. Therefore, it is enough to show $u''\in[\![\varphi\vee\psi]\!]$, which can be concluded from the facts $u''\restriction_Y=u'$ and $u'\in[\![\varphi]\!]$. ◀

▶ **Lemma 24.** *For any finite sets of variables Y,Z and predicates $\varphi\in\Phi_Y$ and $\psi\in\Phi_Z$, if φ and ψ can be arbitrarily over-approximated then $\varphi\wedge\psi$ can be arbitrarily approximated as well.*

Proof. For the purpose of contradiction, suppose $\exists\epsilon\in\mathbb{R}_+\bullet\forall\delta\in\mathbb{R}_+\bullet[\![\varphi^\delta\wedge\psi^\delta]\!]\not\subseteq\mathtt{B}^\epsilon_\infty([\![\varphi\wedge\psi]\!])$. Fix such an ϵ and for any $n\in\mathbb{N}$, let $\epsilon_n:=\frac{\epsilon}{n+1}$, and let $\delta_n\in\mathbb{R}_+$ be a value for which $[\![\varphi^{\delta_n}]\!]\subseteq\mathtt{B}^{\epsilon_n}_\infty([\![\varphi]\!])$ and $[\![\psi^{\delta_n}]\!]\subseteq\mathtt{B}^{\epsilon_n}_\infty([\![\psi]\!])$ hold. Also, make sure for any $n\in\mathbb{N}$, $\delta_{n+1}\leq\delta_n$ and hence $[\![\varphi^{\delta_{n+1}}\wedge\psi^{\delta_{n+1}}]\!]\subseteq[\![\varphi^{\delta_n}\wedge\psi^{\delta_n}]\!]$. Finally, let ν_n be an arbitrary element of the non-empty set $[\![\varphi^{\delta_n}\wedge\psi^{\delta_n}]\!]\setminus\mathtt{B}^\epsilon_\infty([\![\varphi\wedge\psi]\!])$. We know $\nu_n\restriction_Y\in[\![\varphi^{\delta_n}]\!]$ and $\nu_n\restriction_Z\in[\![\psi^{\delta_n}]\!]$.

Using Theorem 2, there is a sequence $m_0<m_1<m_2<\cdots$ such that $\nu_{m_0},\nu_{m_1},\nu_{m_2},\ldots$ converges to ν^*. It is enough to show $\nu^*\in[\![\varphi\wedge\psi]\!]$, because, there are infinitely many indices n such that $\mathtt{d}_\infty(\nu_{m_n},\nu^*)\leq\epsilon$ and any one of them contradicts the fact $\nu_{m_n}\notin\mathtt{B}^\epsilon_\infty([\![\varphi\wedge\psi]\!])$. Note that $\nu_{m_0}\restriction_Y,\nu_{m_1}\restriction_Y,\nu_{m_2}\restriction_Y,\ldots$ converges to $\nu^*\restriction_Y$ and $\nu_{m_0}\restriction_Z,\nu_{m_1}\restriction_Z,\nu_{m_2}\restriction_Z,\ldots$ converges to $\nu^*\restriction_Z$.

By Theorem 11, we know $[\![\varphi]\!]$ is closed and hence $\nu^*\restriction_Y\in[\![\varphi]\!]$ (otherwise, $\nu^*\restriction_Y$ will have a positive distance with $[\![\varphi]\!]$, which is a contradiction). Similarly, we know $\nu^*\restriction_Z\in[\![\psi]\!]$. Therefore, $\nu^*\in[\![\varphi\wedge\psi]\!]$. ◀

▶ **Lemma 25.** *For any finite set of variables X, variable y, interval $\mathtt{I}$, and a predicate $\varphi\in\Phi_X$, if φ can be arbitrarily over-approximated then $\psi:=\exists y\in\mathtt{I}\bullet\varphi$ can be arbitrarily over-approximated as well.*

Proof. The proof is trivial once we noticed for any two points $\nu_1,\nu_2\in\mathbb{R}^X$ we have $\mathtt{d}_\infty(\nu_1\restriction_{X\setminus\{y\}},\nu_2\restriction_{X\setminus\{y\}})\leq\mathtt{d}_\infty(\nu_1,\nu_2)$. ◀

▶ **Lemma 26.** *For any finite set of variables X, variable y, interval $\mathtt{I}$, and a predicate $\varphi\in\Phi_X$, if φ can be arbitrarily over-approximated then $\psi:=\forall y\in\mathtt{I}\bullet\varphi$ can be arbitrarily over-approximated as well.*

Proof. If $\mathtt{I}$ is an empty set then $\forall\delta\in\mathbb{R}\bullet[\![\psi^\delta]\!]=[\![\psi]\!]$. Otherwise, if $y\notin X$ then $\forall\delta\in\mathbb{R}\bullet[\![\psi^\delta]\!]=[\![\varphi^\delta]\!]$. Otherwise, For the purpose of contradiction, suppose $\exists\epsilon\in\mathbb{R}_+\bullet\forall\delta\in\mathbb{R}_+\bullet\exists\nu\in[\![\forall y\in\mathtt{I}\bullet\varphi^\delta]\!]\bullet\nu\notin\mathtt{B}^\epsilon_\infty([\![\forall y\in\mathtt{I}\bullet\varphi]\!])$. Fix such ϵ and let $\nu'\in\mathbb{R}^{\{y\}}$ be an arbitrary point. Also, for every $n\in\mathbb{N}$, let $\nu_n\in[\![\forall y\in\mathtt{I}\bullet\varphi^{\delta_n}]\!]\setminus\mathtt{B}^\epsilon_\infty([\![\forall y\in\mathtt{I}\bullet\varphi]\!])$ be an arbitrary point, where $\delta_n:=\frac{1}{n+1}$.

Using Theorem 2, we know there is a sequence $m_0<m_1<m_2<\ldots$ such that $\nu_{m_0},\nu_{m_1},\nu_{m_2},\ldots$ converges to ν^*. It is enough to show $\nu^*\in[\![\forall y\in\mathtt{I}\bullet\varphi]\!]$, because, there are infinitely many indices n such that $\mathtt{d}_\infty(\nu_{m_n},\nu^*)\leq\epsilon$ and any one of them contradicts the fact $\nu_{m_n}\notin\mathtt{B}^\epsilon_\infty([\![\forall y\in\mathtt{I}\bullet\varphi]\!])$. For any $n\in\mathbb{N}$, we know $\nu_{m_n}\frown\nu'\in[\![\varphi^{\delta_{m_n}}]\!]$. We also know, $\nu_{m_0}\frown\nu',\nu_{m_1}\frown\nu',\nu_{m_2}\frown\nu',\ldots$ converges to $\nu^*\frown\nu'$ which, by Theorem 11 and hence closedness of $[\![\varphi]\!]$, is a point in $[\![\varphi]\!]$. Therefore, $\nu^*\in[\![\forall y\in\mathtt{I}\bullet\varphi]\!]$, since there is no constraint on ν'. ◀