Complete Robust Hybrid Systems Reachability

Noah Abou El Wafa and André Platzer

Karlsruhe Institute of Technology, Karlsruhe, Germany
{noah.abouelwafa,platzer}@kit.edu

Abstract. This paper introduces *robust differential dynamic logic* (a fragment of differential dynamic logic) to specify and reason about *robust hybrid systems*. By small, natural, and practically meaningful syntactic restrictions, specifications are ensured (by construction) to be topologically open and, thus, robust with respect to infinitesimal perturbations without explicit quantitative margins of error in the syntax or in proofs. The main result is a proof of *absolute completeness* of robust differential dynamic logic for reachability properties of general hybrid systems. The proof is constructive, self-contained, and demonstrates how robustly-correct hybrid systems reachability specifications can be automatically verified through proof.

Keywords: Differential dynamic logic · Robust hybrid systems · Hybrid systems reachability

1 Introduction

The study and verification of hybrid systems properties is of fundamental importance for many cyber-physical systems applications. Yet, even reachability is decidable *only* for the most restrictive classes of hybrid systems and the simplest properties [5,1]. The theoretical hardness is often attributed to *excessive precision* that does not reflect physical reality, which in practice always involves some error or imprecision [4]. This has led to a bifurcation of approaches to hybrid systems verification: On the one hand, there are approaches that *weaken the semantics* to regain decidability by admitting positive $\delta > 0$ errors in the decisions, for example δ-decidability [6] or interval approximation [13]. On the other hand, precise and deductive approaches, such as differential dynamic logic [7], retain exact semantics, but rely on *undecidable oracles* [8] or other restrictions [12] for completeness. But neither of these approaches is at the same time free of perturbations or errors *and* complete in general.

This paper bridges the gap, retaining *exact semantics* and nonetheless achieving *completeness*, by focusing on realistic *robust properties* of hybrid systems in a logical calculus. Importantly, robustness is a topological property (openness) that does not require any explicit perturbations, margins of error, or positive disturbances $\delta > 0$. This notion of *infinitesimal robustness* is captured purely syntactically within differential dynamic logic dL [7,10], a highly expressive, powerful logic, which extends first-order dynamic logic with differential equations to

A. Biere et al. (Eds.): IJCAR 2026, LNCS 16689, pp. 342–358, 2026.
DOI: 10.1007/978-3-032-32592-1_20

reason about continuous dynamics. The *robust* fragment (robust differential dynamic logic) syntactically restricts boundary phenomena, which demand unnecessary precision that cannot be delivered symbolically or computationally. This restriction is simple, intuitive and modest, as it demands only that in positive positions (within an even number of negations) all inequalities are strict, and in negative positions (within an odd number of negations) all inequalities are weak. Intuitively, any claim to be proven should be robustly true (i.e., strict or open), while assumptions should conservatively be considered marginal (non-strict or closed). Little of the *practical* expressiveness of dL is lost, since the difference is only between $x > 0$ and $x \geq 0$, which is not practically relevant for the verification of cyber-physical systems. Robust dL provides a clean, simple, syntactic way of describing and reasoning deductively about general robust hybrid systems properties without any need for manual error management.

The *large expressiveness* of dL means that it is impossible to prove all valid formulas in a sound and effective proof calculus [8] or even to decide the truth of all reachability problems formulated in dL (a consequence of Matiyasevich's theorem). It is shown that robustness solves this problem by reducing the actual expressiveness to a completely axiomatizable level without sacrificing practical expressiveness. So robustness *uniformly* handles all difficulties that arise when proving a valid reachability formula. For example, valid existential quantifiers can be correctly instantiated by rational values and unbounded reachability can be bounded by compactness. Unlike some previous completeness theorems [7,8,10], completeness for robust hybrid systems reachability is *not relative* and does not require an undecidable oracle: any true robust hybrid systems reachability property is provable unconditionally. Importantly, reachability properties are not time-bounded, but can involve hybrid systems running for unbounded lengths of time and repeating loops for an arbitrary number of iterations.

Contributions. First, *robust differential dynamic logic* is introduced as a complete fragment of dL capable of expressing all robust properties of hybrid systems. It is shown that the semantics naturally reflects this robustness through a topological characterization (Section 2). Second, a simple iterative axiomatization of ODE properties is presented (Section 3). Third, robustness is used to provide a constructive completeness proof for robust hybrid systems reachability properties. This *non-relative completeness proof* for a general class of hybrid systems properties overcomes the previous reliance on undecidable oracles and restrictions to algebraic hybrid systems (Section 4).

Related Work. Hybrid systems verification is undecidable [5]. A δ-decidability algorithm handles logical reachability analysis via SMT solving approximately for accuracy $\delta > 0$ [6]. Similar syntactic perturbations have been used to prove quasi-decidability of safety for hybrid systems [13]. Reachability for restricted robust systems by approximation has been proved decidable [4]. Unlike these approaches, robust differential dynamic logic handles robustness in a simple and topological way, without requiring explicit approximations.

Completeness of dL relative to its continuous fragment and its discrete fragment has been shown [8]. Absolute completeness was established for algebraic properties of algebraic hybrid systems [12] and invariance properties of hybrid systems [12,16]. This paper shows *absolute* completeness for robust reachability properties of *general hybrid systems*, extending the completeness of open first-order properties of purely continuous dynamical systems [11].

Conceptually related is the proof of completeness of the uninterpreted diamond fragment of first-order dynamic logic [15]. The connection between open sets of reals and decidability is fundamental in computable analysis [19].

2 Differential Dynamic Logic and Robustness

This section first recalls the syntax and semantics of differential dynamic logic dL [7] (Sections 2.1 and 2.2). Robust differential dynamic logic dL_r and its reachability fragment $\mathsf{dL}_{\mathsf{r}\diamond}$ are then defined as syntactic fragments of dL (Sections 2.3 and 2.4). The semantics of $\mathsf{dL}_{\mathsf{r}\diamond}$ are shown to be robust under infinitesimal perturbations and thus capture the *robustly true* formulas (Section 2.5).

2.1 Syntax of Differential Dynamic Logic

Formally, dL extends first-order dynamic logic for (discrete) programs, with continuous programs to model dynamics of hybrid systems. As dL is interpreted over the real numbers, the terms of dL are ordinary terms of real arithmetic:

$$v, w ::= x \mid q \mid v + w \mid v \cdot w \qquad \text{dL terms}$$

where $x \in \mathcal{V}$ is a first-order variable from some fixed set $\mathcal{V}$ of variables and q is a rational constant. Terms are essentially syntactic descriptions of polynomials in the variables $\mathcal{V}$ with rational coefficients ($\mathbb{Q}[\mathcal{V}]$). Symbols x and v implicitly range over *tuples* of variables $x_1, \ldots, x_n$ and terms $v_1, \ldots, v_n$, respectively.

Formulas φ and hybrid programs α of dL are described by:

$$\varphi, \rho ::= v > w \mid \varphi \vee \rho \mid \exists x\, \varphi \mid \langle \alpha \rangle \varphi \qquad \text{dL formulas}$$
$$\alpha, \gamma ::= x{:=}v \mid ?\varphi \mid \alpha \cup \gamma \mid \alpha; \gamma \mid \alpha^* \mid \mathsf{x}' = \mathsf{v}\,\&\,\varphi \qquad \text{dL programs}$$

where x and v are tuples (of the same length) of variables and terms, respectively, so that $\mathsf{x}' = \mathsf{v}\,\&\,\varphi$ describes the symbolic ordinary differential equation $x_1' = v_1, \ldots, x_n' = v_n$ on the domain defined by the formula φ.

For convenience it is assumed, without loss of generality, that there is a special variable τ and every continuous evolution modality contains $\tau' = 1$. (This does not change the proof theory by the differential ghost axiom [12].)

Interpretation of dL. The modalities $\langle \alpha \rangle \varphi$ of dL are parametrized by *hybrid programs* α, which describe hybrid systems *syntactically*. A formula of the form $\langle \alpha \rangle \varphi$ asserts the existence of an execution of the hybrid program α that reaches

a state in which φ is true. A hybrid program α describes a non-deterministic hybrid system. The *deterministic assignment* $x{:=}v$ models a simple state change, where the value of the variable x is updated to the value of the term v. The *test program* $?\varphi$ proceeds without effect as long as φ is true in the current state and aborts otherwise. The *non-deterministic choice program* $\alpha \cup \gamma$ is a non-deterministic (existential) branch: the evolution may continue according to one of the two hybrid programs α or γ. A *sequential composition* $\alpha;\gamma$ of hybrid programs executes γ after α. For finite iteration, i.e., for $\alpha;\ldots;\alpha$ repeated n times, write α^n and let $\alpha^{\leq n} \equiv ?\text{true} \cup \alpha^1 \cup \alpha^2 \cup \ldots \cup \alpha^n$ stand for bounded non-deterministic repetition. The *iteration program* α^* represents finite repetition of α for an arbitrary (non-deterministic) number of times. Finally, the *continuous program* $\mathsf{x}' = \mathsf{v} \,\&\, \varphi$ describes the continuous evolution of the state along the solution of the differential equation for a non-deterministic amount of time, as long as φ, called the *evolution domain*, is satisfied.

Notation and Abbreviations. As usual, $\geq$, $=$, $\rightarrow$ and the dual connectives are defined via negation. Moreover, the box modalities $[\alpha]\varphi$ abbreviate $\neg\langle\alpha\rangle\neg\varphi$ to say that every execution of α ends in a state in which φ is true. Syntactically, for a tuple of terms v, the ∞-norm is used via the abbreviation

$$|\mathsf{v}| < w \;\equiv\; -w < v < w \wedge \ldots \wedge -w < v < w$$

and similarly for $|\mathsf{v}| \sim w$ where $\sim \in \{\leq, >, \geq\}$.

Free Variables. The free variables of a formula are, as usual, the variables x that appear somewhere in φ not within the scope of a quantifier or assignment that *necessarily* binds x. For example, x is free in $[y{:=}x]y > 0$ and y is not free. (A formal definition is in the literature [9, Definition 8].) The set of free variables of φ is denoted $\mathrm{FV}(\varphi)$. A *sentence* is a formula without free variables.

2.2 Semantics of Differential Dynamic Logic

The semantics of differential dynamic logic intuitively describes the intended behaviour of hybrid systems models. First basic definitions and notations are introduced, before the denotational semantics of dL [7] is recalled.

Reachability Relations. Hybrid systems behaviour is semantically described by a *reachability relation* R that describes which states can be reached from a given initial state. The set $R^{-1}(U)$ of states from which a target set $U \subseteq X$ of states is reachable along a relation $R \subseteq X \times X$ is defined by $R^{-1}(U) = \{x : \exists y \in U\ (x,y) \in R\}$. The composition of two relations R, S is $R \circ S = \{(x,z) : \exists y\ (x,y) \in R, (y,z) \in S\}$ and $(R \circ S)^{-1}(U) = R^{-1}(S^{-1}(U))$. The n-fold composition of R with itself is written $R^n = R \circ \ldots \circ R$. The *reflexive transitive closure* of a relation R is $R^* = \bigcup_{n \geq 0} R^n$.

Differential Equations. The continuous part of a hybrid system is usually described by an ordinary differential equation modeling the dynamics of motion. In this paper, hybrid systems are specified symbolically, and attention is restricted to *polynomial differential equations* $x' = f(x)$ where $f : \mathbb{R}^n \to \mathbb{R}^n$ is a polynomial with rational coefficients.

By the Picard-Lindelöf theorem [18], rational polynomial differential equations locally have unique solutions for every initial value $x_0 \in \mathbb{R}^n$ and every solution extends to a solution on an open maximal interval of existence. Let $I_{x_0} \subseteq \mathbb{R}$ denote this maximal interval for the initial value problem $x' = f(x)$, $x(0) = x_0$ and let $\Phi_f(x_0, \cdot) : I_{x_0} \to \mathbb{R}^n$ denote its unique solution. The partial function $\Phi_f : \Omega_f = \{(x_0, t) \in \mathbb{R}^n \times \mathbb{R} : t \in I_{x_0}\} \to \mathbb{R}^n$ is the flow of the differential equation. The flow is *continuous* on the domain, and if the coefficients of the polynomial f_α vary continuously in α, then $\Phi_{f_\alpha}(x_0, t)$ is continuous in α, x_0 and t. Proofs of these facts can be found in the literature [18].

Denotational Semantics The semantics of dL is defined in the literature [8] and recalled here for completeness. A *state* is a function $\omega : \mathcal{V} \to \mathbb{R}$ from the set of variables $\mathcal{V}$ to real numbers, which assigns a value $\omega(x)$ to every variable x and $\mathcal{S} = \mathbb{R}^{\mathcal{V}}$ is the set of *all* states equipped with the product topology. For a tuple of variables x and a tuple a of values of the same length, $\omega^{\mathsf{a}}_{\mathsf{x}}$ is the state that coincides with ω everywhere, except $\omega^{\mathsf{a}}_{\mathsf{x}}(x_i) = a_i$ for $x_i \in \mathsf{x}$. The restriction of a state ω to a tuple x of variables is the tuple $\omega\restriction_{\mathsf{x}} = (\omega(x_1), \ldots, \omega(x_n))$.

The semantics of a term with respect to a state $\omega[\![v]\!] \in \mathbb{R}$ is a real number:

$$\omega[\![x]\!] = \omega(x) \quad \omega[\![q]\!] = q \quad \omega[\![v + w]\!] = \omega[\![v]\!] + \omega[\![w]\!] \quad \omega[\![v \cdot w]\!] = \omega[\![v]\!] \cdot \omega[\![w]\!]$$

The semantics of a formula φ is the set $[\![\varphi]\!] \subseteq \mathcal{S}$ of states which satisfy φ and the semantics of a hybrid program α is the reachability relation $[\![\alpha]\!] \subseteq \mathcal{S} \times \mathcal{S}$ described by the program:

$$\begin{array}{ll}
[\![v > w]\!] = \{\omega : \omega[\![v]\!] > \omega[\![w]\!]\} & [\![\varphi \vee \rho]\!] = [\![\varphi]\!] \cup [\![\rho]\!] \\
[\![\exists x\, \varphi]\!] = \{\omega : \exists a \in \mathbb{R}\ \omega^a_x \in [\![\varphi]\!]\} & [\![\langle\alpha\rangle\varphi]\!] = [\![\alpha]\!]^{-1}([\![\varphi]\!]) \\
[\![x{:=}v]\!] = \{(\omega, \omega^{\omega[\![v]\!]}_x) : \omega \in \mathcal{S}\} & [\![?\varphi]\!] = \{(\omega, \omega) : \omega \in [\![\varphi]\!]\} \quad [\![\alpha^*]\!] = [\![\alpha]\!]^* \\
[\![\alpha \cup \gamma]\!] = [\![\alpha]\!] \cup [\![\gamma]\!] & [\![\alpha; \gamma]\!] = [\![\alpha]\!] \circ [\![\gamma]\!]
\end{array}$$

For a continuous program $\mathsf{x}' = \mathsf{v}$ with n variables and a state ω, the polynomial vector field $f_\omega : \mathbb{R}^n \to \mathbb{R}^n$ is defined by $f_\omega(\mathsf{a}) = \omega^{\mathsf{a}}_{\mathsf{x}}[\![\mathsf{v}]\!]\restriction_{\mathsf{x}}$, and the semantics of a continuous program is

$$\begin{aligned}
[\![\mathsf{x}' = \mathsf{v}\,\&\,\varphi]\!] = &\{(\omega, \Psi_\omega(t)) : t \geq 0\,,\ (\omega, t) \in \Omega_{f_\omega}\,,\ \Psi_\omega([0,t]) \subseteq [\![\varphi]\!]\} \\
&\text{where } \Psi_\omega(t) = \omega^{\Phi_{f_\omega}(\omega\restriction_{\mathsf{x}}, t)}_{\mathsf{x}} \text{ is the state-flow.}
\end{aligned}$$

As usual, a formula φ is said to be *valid* if $[\![\varphi]\!] = \mathcal{S}$.

2.3 Robust Differential Dynamic Logic

Robust differential dynamic logic (dL_r) is a careful refinement of dL for the purpose of *syntactically* ensuring that any formula describes properties with locally robust truth. An example of such a formula is $v > w$. If it is true in some interpretation of the variables, then it remains true under a sufficiently small perturbation of the variables. However, the formula $-x^2 \geq x^2$ is *not* robust, as it is true only in the *razor-edge case* when x is interpreted as 0, and it is false for any other interpretation of x, no matter how close.

Robust dL generalizes the distinction between strict inequality ($>$) and weak inequality ($\geq$) from real numbers to properties of hybrid systems generally. There are two kinds of dL_r formulas: *strict formulas* φ° generalizing strict inequality and *non-strict formulas* $\varphi^\bullet$ generalizing weak inequality, which are described by:

$$\varphi^\circ, \rho^\circ ::= v > w \mid \varphi^\circ \vee \rho^\circ \mid \varphi^\circ \wedge \rho^\circ \mid \exists x\, \varphi^\circ \mid \forall x\, \varphi^\circ \mid \langle \alpha^\circ \rangle \varphi^\circ \mid [\alpha^\bullet]\varphi^\circ \qquad \text{strict } \mathsf{dL}_\mathsf{r}$$
$$\varphi^\bullet, \rho^\bullet ::= v \geq w \mid \varphi^\bullet \vee \rho^\bullet \mid \varphi^\bullet \wedge \rho^\bullet \mid \exists x\, \varphi^\bullet \mid \forall x\, \varphi^\bullet \mid \langle \alpha^\bullet \rangle \varphi^\bullet \mid [\alpha^\circ]\varphi^\bullet \qquad \text{non-strict } \mathsf{dL}_\mathsf{r}$$

The box and diamond modalities of dL_r formulas are again described by two different classes of hybrid programs. Corresponding to the difference between strict and non-strict formulas, there are *strict* hybrid programs α° and *non-strict* hybrid programs $\alpha^\bullet$ defined by

$$\alpha^\circ, \gamma^\circ ::= x{:=}v \mid ?\varphi^\circ \mid \alpha^\circ \cup \gamma^\circ \mid \alpha^\circ; \gamma^\circ \mid \alpha^{\circ *} \mid \mathsf{x}' = \mathsf{v} \,\&\, \varphi^\circ \qquad \text{strict } \mathsf{dL}_\mathsf{r} \text{ program}$$
$$\alpha^\bullet, \gamma^\bullet ::= x{:=}v \mid ?\varphi^\bullet \mid \alpha^\bullet \cup \gamma^\bullet \mid \alpha^\bullet; \gamma^\bullet \mid \alpha^{\bullet *} \mid \mathsf{x}' = \mathsf{v} \,\&\, \varphi^\bullet \qquad \text{non-strict } \mathsf{dL}_\mathsf{r} \text{ program}$$

The difference between strict and non-strict programs is that tests and evolution domains are strict formulas in strict programs and non-strict formulas in non-strict programs, respectively. Note that in a strict formula the programs appearing in a diamond modality need to be strict programs, while the programs in a box modality need to be non-strict programs. Dually, the diamond programs in a non-strict formula are non-strict programs, while the box programs in a non-strict formula are strict programs.

By design, robust differential dynamic logic consists of two *fragments* of differential dynamic logic, since φ° and $\varphi^\bullet$ *are* formulas of differential dynamic logic. The circle annotations are merely for emphasis, indicating that a formula belongs to the strict (open $\circ$) or the non-strict (closed $\bullet$) fragment. In fact, the strict (non-strict) formulas of dL_r correspond exactly to the fragment of dL in which strict (weak) inequalities appear only in positions of positive polarity and weak (strict) inequalities appear only in positions of negative polarity. (For a full definition of polarity in dL, see Section A.) From a modeling and specification perspective, therefore, little practical expressiveness is lost, since the distinction between $x \geq 0$ and $x > 0$ is not of practical significance. For example, hybrid systems controllers with equality constraints $?x = v$ may not appear in strict programs. This restriction is reasonable, as a controller cannot *exactly verify* that the value of the real variable x is 0, which would require checking infinitely many decimal places of x [19]. Instead, a controller model may, more realistically, test $?(v - 0.1 < x < v + 0.1)$.

Negation of Robust Formulas. Although negation is not a syntactic primitive of $\mathsf{dL_r}$, it bridges the strict and the non-strict fragments: The negation $\neg(\rho^\circ)$ of a strict formula is a non-strict formula and conversely the negation $\neg(\varphi^\bullet)$ of a non-strict formula is a strict formula. For instance, the implication $\varphi^\bullet \to \rho^\circ$ is a strict formula $\neg(\varphi^\bullet) \vee \rho^\circ$ when $\varphi^\bullet$ is a non-strict formula and ρ° is a strict formula. Dually, the implication $\rho^\circ \to \varphi^\bullet$ is a non-strict formula $\neg(\rho^\circ) \vee \varphi^\bullet$.

Robust Negation. As noted, the negation of a strict robust formula is not a strict robust formula. *Robust negation* is a slightly stronger notion defined by

$$\sim(v > w) \equiv w > v \qquad\qquad \sim(v \geq w) \equiv w \geq v$$

and extended to formulas homomorphically. The robust negation $\sim\varphi^\circ$ of a strict (non-strict) $\mathsf{dL_r}$ formula φ° is a strict (non-strict) $\mathsf{dL_r}$ formula.

Robust negation of $\mathsf{dL_r}$ formulas is a strengthening of negation that does not satisfy the law of the excluded middle. That is, the robust negation $\sim\varphi^\circ$ entails the standard negation $\neg\varphi^\circ$, but not conversely. A strict $\mathsf{dL_r}$ formula may be viewed as having one of three truth values in a state: φ can be *robustly true*, if it is semantically satisfied, *robustly false* if $\sim\varphi$ is satisfied and *undetermined* if neither is the case. For example, the formula $x > 0$ is robustly true in a state with $\omega(x) = 1$ and robustly false in a state with $\omega(x) = -1$. However, in a state with $\omega(x) = 0$, the truth value of $x > 0$ is not *robustly determined*. A $\mathsf{dL_r}$ formula φ is said to be *robustly determined* if $\varphi \vee \sim\varphi$ is valid.

Bounded Quantification. Bounded quantifiers are important in the definition of the reachability fragment of $\mathsf{dL_r}$. For a strict formula φ° and a term v, which does not mention x, the formula $\exists x{>}v\, \varphi^\circ$ stands for the strict formula $\exists x\, (x > v \wedge \varphi^\circ)$. Dually, $\forall x{\geq}v\, \varphi^\circ$ abbreviates the strict formula $\forall x\, (x \geq v \to \varphi^\circ)$. To exclude ambiguous notation such as $\exists x{>}x\, \varphi^\circ$, it is important that the term v may not mention x. For a closed interval $I = [v, w]$ of terms v, w not mentioning x, the strict formula $\forall_I x\ \varphi^\circ$ and the non-strict formula $\exists_I x\ \varphi^\bullet$ are defined by

$$\forall_I x\ \varphi^\circ \equiv \forall x\, ((v \leq x \wedge x \leq w) \to \varphi^\circ) \qquad \exists_I x\ \varphi^\bullet \equiv \exists x\, (v \leq x \wedge x \leq w \wedge \varphi^\bullet)$$

The strict formula $\exists_I x\ \varphi^\circ$ and the non-strict formula $\forall_I x\ \varphi^\bullet$ are defined analogously for the open interval $I = (v, w)$.

2.4 Robust Reachability Fragment

Hybrid systems reachability properties are expressed using *diamond modalities*. Robust reachability differential dynamic logic $\mathsf{dL_{r\diamond}}$ is a fragment of $\mathsf{dL_r}$ that contains arbitrary robust diamond modalities. This makes it expressively powerful and at the same time tame enough to be complete, as Section 4 will show.

Strict Robust Reachability. Syntactically, the *strict robust reachability* formulas and programs of robust reachability differential dynamic logic $\mathsf{dL}_{\mathsf{r}\diamond}$ are:

$$\varphi^\diamond, \rho^\diamond ::= v > w \mid \varphi^\diamond \vee \rho^\diamond \mid \varphi^\diamond \wedge \rho^\diamond \mid \exists x\, \varphi^\diamond \mid \forall_I x\, \varphi^\diamond \mid \langle \alpha^\diamond \rangle \varphi^\diamond \mid [\alpha^\blacklozenge]\varphi^\diamond \quad \text{strict } \mathsf{dL}_{\mathsf{r}\diamond}$$
$$\alpha^\diamond, \gamma^\diamond ::= x{:=}v \mid ?\varphi^\diamond \mid \alpha^\diamond \cup \gamma^\diamond \mid \alpha^\diamond; \gamma^\diamond \mid \alpha^{\diamond *} \mid \mathsf{x}' = \mathsf{v} \,\&\, \varphi^\diamond \quad \text{strict } \mathsf{dL}_{\mathsf{r}\diamond}$$

where $I = [v, w]$ is an interval of terms, which do not mention x, and $\alpha^\blacklozenge$ is a non-strict robust reachability program (see below). Importantly, strict robust reachability formulas capture essentially *all* physically relevant reachability (diamond) properties, including unbounded continuous evolution and loop iteration. The restriction that universal quantifiers are bounded means only that a $\mathsf{dL}_{\mathsf{r}\diamond}$ formula cannot *assume* a variable to have *any* arbitrary value. For specification, it should always be possible to find an appropriate bounding range. Moreover, strict $\mathsf{dL}_{\mathsf{r}\diamond}$ does not restrict the reachability properties, which may feature *unbounded* loops and unbounded continuous evolution.

Non-strict Robust Reachability. Although the focus of $\mathsf{dL}_{\mathsf{r}\diamond}$ is on reachability formulas, which need only *diamond* modalities, the syntax of strict $\mathsf{dL}_{\mathsf{r}\diamond}$ formulas admits a restricted set of safety formulas (box modalities) involving non-strict formulas and programs of $\mathsf{dL}_{\mathsf{r}\diamond}$. Non-strict $\mathsf{dL}_{\mathsf{r}\diamond}$ formulas do not allow iterations and all continuous evolutions need to be bounded (in space and time).

$$\varphi^\blacklozenge, \rho^\blacklozenge ::= v \geq w \mid \varphi^\blacklozenge \vee \rho^\blacklozenge \mid \varphi^\blacklozenge \wedge \rho^\blacklozenge \mid \exists_I x\, \varphi^\blacklozenge \mid \forall x\, \varphi^\blacklozenge \mid \langle \alpha^\blacklozenge \rangle \varphi^\blacklozenge \mid [\alpha^\diamond]\varphi^\blacklozenge \quad \text{non-strict } \mathsf{dL}_{\mathsf{r}\diamond}$$
$$\alpha^\blacklozenge, \gamma^\blacklozenge ::= x{:=}v \mid ?\varphi^\blacklozenge \mid \alpha^\blacklozenge \cup \gamma^\blacklozenge \mid \alpha^\blacklozenge; \gamma^\blacklozenge \mid \mathsf{x}' = \mathsf{v} \,\&\, \varphi^\blacklozenge \wedge |\mathsf{x}| \leq k \quad \text{non-strict } \mathsf{dL}_{\mathsf{r}\diamond}$$

The negation of a strict $\mathsf{dL}_{\mathsf{r}\diamond}$ formula is a non-strict $\mathsf{dL}_{\mathsf{r}\diamond}$ formula and vice versa. This can be seen inductively as, for example, the negation $\langle \alpha^\blacklozenge \rangle \neg \varphi^\diamond$ of the strict formula $[\alpha^\blacklozenge]\varphi^\diamond$ is a non-strict formula. The annotations $\diamond$ and $\blacklozenge$ indicate that a formula belongs to one of the *reachability* fragments consisting of strict and non-strict formulas and programs of $\mathsf{dL}_{\mathsf{r}\diamond}$, respectively.

The definition of non-strict $\mathsf{dL}_{\mathsf{r}\diamond}$ programs ensures that all box modalities in strict $\mathsf{dL}_{\mathsf{r}\diamond}$ formulas are iteration-free and have bounded continuous evolution. To see why this is necessary, observe that loops in box-modalities can encode universal quantifiers over the natural numbers $\forall x \in \mathbb{N}\, \varphi$ via $[x{:=}0; (x{:=}x+1)^*]\varphi$. By Matiyasevich's theorem, the validity of such formulas cannot be semi-decidable and consequently there can be no complete proof calculus for strict $\mathsf{dL}_{\mathsf{r}\diamond}$ formulas, if loops in box modalities are allowed. For this reason, the restriction to the strict *reachability fragment*, where iteration programs may only appear in diamond formulas, is critical.

The assumption that the evolution of continuous programs in a box modality is bounded $|\mathsf{x}| \leq k$ mirrors the boundedness restriction on universal quantifiers. Both ensure that universal properties are restricted to a *pre-defined range*. The evolution bound also enforces an upper bound on the duration of the evolution, since the differential equation is assumed to include a time variable $\tau' = 1$. All the restrictions involved in the robust reachability fragment are, to some extent, necessary to ensure that the fragment is natural and a complete calculus exists.

2.5 Robustness Semantically

The crucial semantic property of $\mathsf{dL}_{\mathsf{r}\diamond}$ is that the semantics topologically reflects the robustness property which was enforced syntactically. Assumptions should be *topologically closed* and conclusions should be *open*. This is subtle, as it relies on continuity of flows of differential equations and needs to deal *topologically* with infinite unions (existential quantifiers) and infinite intersections (universal quantifiers). The key is a compactness argument on the *domain of universal quantification*, which relies on a standard topological lemma.

Lemma 1 (Tube Lemma). *For X, Y topological spaces, Y compact and $U \subseteq X \times Y$ open in the product topology, the set $\{x \in X : \forall y \in Y\ (x, y) \in U\}$ is open.*

See proof on page 361. By induction it can be shown that strict reachability formulas are semantically open and non-strict reachability formulas are closed. Lemma 1 ensures that bounded universal quantification preserves these properties.

Theorem 2 (Semantic Robustness). *For $\mathsf{dL}_{\mathsf{r}\diamond}$ formulas and programs:*

1. $[\![\varphi^\diamond]\!]$ *is open for strict formulas* $\varphi^\diamond$
2. $[\![\varphi^\bullet]\!]$ *is closed for non-strict formulas* $\varphi^\bullet$
3. $[\![\alpha^\diamond]\!]^{-1}(U)$ *is open for strict programs* $\alpha^\diamond$ *and open sets* U
4. $[\![\alpha^\bullet]\!]^{-1}(C)$ *is closed for non-strict programs* $\alpha^\bullet$ *and closed sets* C

See proof on page 362. The defining restriction of $\mathsf{dL}_{\mathsf{r}\diamond}$ formulas that loops may not appear in box modalities is crucial for (4). The proof of (4) uses only time-boundedness. Instead of assuming $|\mathsf{x}| \leq k$ in the evolution domain, it suffices to assume $\tau \leq k$. The proof goes through when allowing non-strict programs $\mathsf{x}' = \mathsf{v} \,\&\, \varphi^\bullet \wedge \tau \leq k$.

3 Axioms and Rules of dL

The power of differential dynamic logic comes from its proof calculus, which is relatively complete [8] and can prove all invariants and all algebraic properties of algebraic hybrid systems [12]. Moreover, all true open properties of compact initial value problems are provable [11]. In Section 4, completeness for general robust reachability properties will be shown. The proof calculus for dL can be found in the literature [8,12]. The relevant (derived) rules and axioms for the completeness proof are recalled. Soundness for robust dL is inherited from soundness for dL, since dL_{r} is a syntactic fragment and their semantics coincide.

3.1 Basic dL Calculus

The dL calculus is a generalization of classical first-order sequent calculus and the calculus of first-order dynamic logic. A sequent $\Gamma \vdash \Delta$ consists of two (unordered) sets Γ, Δ of formulas. As usual, the left-hand side of the sequent is read conjunctively and the right-hand side disjunctively. The usual sequent comma notation, as in $\varphi, \Gamma_1, \Gamma_2 = \{\varphi\} \cup \Gamma_1 \cup \Gamma_2$, is used.

Real Arithmetic Foundation. The foundational level of differential dynamic logic is the theory of real arithmetic. Because it is decidable [17], real arithmetic is simply taken as the basic rule on which the proof calculus is built. The rule

$$\mathbb{R}\ \frac{*}{\Delta \vdash \Gamma}\quad (\text{if } \textstyle\bigwedge_{\rho\in\Delta}\rho \to \bigvee_{\varphi\in\Delta}\varphi \text{ is valid in } \mathbb{R})$$

discharges any sequent in a proof tree that is semantically valid in $\mathbb{R}$, i.e., a tautology of real arithmetic. The interesting and difficult parts of hybrid systems verification, therefore, arise from hybrid programs, in particular *unbounded iteration* and *continuous evolution*. The remaining real arithmetic problems can be handled algorithmically by real arithmetic SMT solving [2].

Basic Logical Proof Rules. The basic logical rules for the standard connectives are the usual sequent calculus rules. Only some of the rules are needed for the completeness proof, and these are recalled here in adapted (derivable) form:

$$\vee\mathrm{R}\ \frac{\Delta \vdash \varphi, \rho, \Gamma}{\Delta \vdash \varphi \vee \rho, \Gamma} \qquad \wedge\mathrm{R}\ \frac{\Delta \vdash \varphi, \Gamma \quad \Delta \vdash \rho, \Gamma}{\Delta \vdash \varphi \wedge \rho, \Gamma}$$

$$\exists\mathrm{R}\ \frac{\Delta \vdash \langle x{:=}v\rangle\varphi, \Gamma}{\Delta \vdash \exists x\, \varphi, \Gamma} \qquad \forall\mathrm{R}\ \frac{\Delta, x \in I \vdash \varphi, \Gamma}{\Delta \vdash \forall_I x\, \varphi, \Gamma}\quad (x \notin \mathrm{FV}(\Delta, \Gamma))$$

$$\mathrm{C}\ \frac{\vdash \varphi \leftrightarrow \rho \quad \Delta \vdash \rho, \Gamma}{\Delta \vdash \varphi, \Gamma}$$

The rules $\vee$R, $\wedge$R and $\forall$R are standard. The existence axiom $\exists$R is usually formulated in terms of substitution of v for x in φ. The formulation in terms of the assignment modality is used only for the convenience of concentrating all subtleties related to substitution in the single case of assignment programs. Recall that x does not appear in the bounds of the interval I in axiom $\forall$R. To apply $\forall$R in a context where x is free in Γ or Δ, it is always possible to *uniformly rename all* occurrences of x in φ before applying $\forall$R. Rule C is a simplified form of the cut rule that is used to instantiate the hybrid systems axioms below.

Assignment Programs and Substitutions. Assignment hybrid programs are usually axiomatized by an axiom of the form $\langle x{:=}v\rangle\varphi \leftrightarrow \varphi_x^v$, where φ_x^v denotes the substitution of v for x in φ. However, this causes unnecessary difficulties with free variable capture and inadmissible substitutions. When x is both free and bound, as for example in a repetition program $(x{:=}x+1)^*$, substitution is subtle. To simplify this, an alternative axiomatization based on the equivalence

$$\langle x{:=}v\rangle\varphi \leftrightarrow \forall z\,(z = x \to \forall x\,(x = v_x^z \to \varphi))$$

is used, where z is suitably fresh and v_x^z is the term v with x replaced uniformly by z. This avoids substitution into formulas by universally quantifying over x and ensuring that it takes the correct value. In case v mentions x, the value of x is retained in the scope of the universal quantifier $\forall x$ by means of the fresh variable z. For convenience, write $\varphi_x^v \equiv \forall z\,(z = x \to \forall x\,(x = v_x^z \to \varphi))$.

Hybrid Program Axioms. In order to handle hybrid programs, dL axiomatizes the hybrid program connectives exactly as dynamic logic:

$$\langle{:=}\rangle \quad \langle x{:=}v\rangle\varphi \leftrightarrow \varphi\tfrac{v}{x} \quad (z \notin \mathrm{FV}(v,\varphi,x)) \qquad \langle ?\rangle \quad \langle ?\rho\rangle\varphi \leftrightarrow (\rho \wedge \varphi)$$

$$\langle\cup\rangle \quad \langle\alpha\cup\gamma\rangle\varphi \leftrightarrow \langle\alpha\rangle\varphi \vee \langle\gamma\rangle\varphi \qquad \langle;\rangle \quad \langle\alpha;\gamma\rangle\varphi \leftrightarrow \langle\alpha\rangle\langle\gamma\rangle\varphi$$

$$\langle*\rangle \quad \langle\alpha^*\rangle\varphi \leftrightarrow \varphi \vee \langle\alpha\rangle\langle\alpha^*\rangle\varphi \qquad \langle^{\leq n}\rangle \quad \frac{\Delta \vdash \langle\alpha^{\leq n}\rangle\varphi, \Gamma}{\Delta \vdash \langle\alpha^*\rangle\varphi, \Gamma}$$

Axiom $\langle{:=}\rangle$ is the *substitution-safe* version of the usual assignment axiom and axioms $\langle ?\rangle$, $\langle\cup\rangle$ and $\langle;\rangle$ reduce program operations to logical connectives equivalently. Axiom $\langle*\rangle$ characterizes loop reachability and the derivable rule $\langle^{\leq n}\rangle$ says that in case φ can be reached in at most n-steps, then an iteration of some length can reach φ.

3.2 Continuous Program Axioms

Continuous programs require a few additional axioms, such as dI for differential invariance reasoning [12], differential cut dC to accumulate information, Δ for relative completeness [8] and real induction axioms [12] to globalize local arguments. This section presents the axioms used in the $\mathsf{dL}_{\mathsf{r}\diamond}$ completeness proof.

Continuous Safety-Reachability Duality. An important insight for the completeness proof is a robust duality between continuous reachability and continuous safety. *Robust* reachability and safety (box) properties of continuous programs are *inter-translatable*. Intuitively, an evolution is safe for bounded time and within a bounded domain if and only if it reaches the time bound or leaves the domain before leaving the safe region.

Theorem 3 (ODE Duality Soundness). *The ODE duality axiom is sound*

$$[\mathrm{ODE}\rangle \quad [\mathsf{x}' = \mathsf{v}\,\&\,\rho^\bullet \wedge |\mathsf{x}| \leq k]\varphi^\diamond \leftrightarrow \langle \mathsf{x}' = \mathsf{v}\,\&\,\varphi^\diamond \vee |\mathsf{x}| > k\rangle(\neg\rho^\bullet \vee |\mathsf{x}| > k)$$

for a strict $\mathsf{dL}_{\mathsf{r}\diamond}$ *formula* $\varphi^\diamond$*, a non-strict* $\mathsf{dL}_{\mathsf{r}\diamond}$ *formula* $\rho^\bullet$ *and a term* k *not mentioning variables from* x*.*

See proof on page 362. The duality axiom is particular to $\mathsf{dL}_{\mathsf{r}\diamond}$. As it relies on robustness, it does not hold for dL formulas, as the following incorrect instance with the strict evolution domain formula $\rho \equiv x < 0$ demonstrates:

$$[\mathsf{x}' = \mathsf{v}\,\&\,x < 0 \wedge |\mathsf{x}| \leq 1]x < 0 \leftrightarrow \langle \mathsf{x}' = \mathsf{v}\,\&\,x < 0 \vee |\mathsf{x}| > 1\rangle(x \geq 0 \vee |\mathsf{x}| > 1)$$

The topological properties of the semantics of $\mathsf{dL}_{\mathsf{r}\diamond}$ formulas prevent such gaps.

Soundness of the axiom $[\mathrm{ODE}\rangle$ crucially relies on the space-boundedness and the time-boundedness of the differential equation. Time-boundedness is ensured by $|\mathsf{x}| \leq k$ as a time component $\tau' = 1$ is always included in $\mathsf{x}' = \mathsf{v}$. The space-boundedness is important, as it makes it possible to handle solutions which exist only for finite time and therefore leave the bounded domain.

Differential Equations via Iterations. An important ingredient to handle the continuous parts of hybrid systems is a reduction of the continuous dynamics to discrete dynamics by means of an Euler approximation similar to axiom $\overleftrightarrow{\Delta}$ [8]. This is possible because robustness ensures that if a state satisfying φ is reachable, then there is some margin of error $\varepsilon > 0$ and a reachable state ω, such that φ is true everywhere in the ε-neighborhood for ω. Thus, continuous reachability can be rephrased by reachability along a discrete Euler approximation of error at most ε. This *quantitative robustness* is syntactically described by the *ε-interior* $\varphi_{\mathsf{x}\pm\varepsilon}$ of a strict formula φ:

$$\varphi_{\mathsf{x}\pm\varepsilon} \equiv \forall_{[x_1-\varepsilon,x_1+\varepsilon]} y_1 \ \ldots \forall_{[x_n-\varepsilon,x_n+\varepsilon]} y_n \ \varphi^{\mathsf{y}}_{\mathsf{x}}$$

which requires φ to be true in a neighborhood of x.

To symbolically bound the error of the Euler approximation of a differential equation $x' = f(x)$ on a compact set, bounds on f and the Jacobian Df, respectively, are required. For a syntactic differential equation $\mathsf{x}' = \mathsf{v}$ let $\frac{\partial v_i}{\partial x_j}$ denote the formal partial derivative of the polynomial term v_i with respect to variable x_j. The formula $\text{bound}^\diamond \equiv \forall|\mathsf{x}|{\leq}k+2\,(|v| < m \wedge \bigwedge_{i,j} |\frac{\partial v_i}{\partial x_j}| < \ell)$ ensures that the syntactic vector field is bounded by m and its Jacobian is bounded by ℓ in the ∞-norm on the set defined by $|\mathsf{x}| \leq k+2$.

The Euler-step program $\text{step}^\diamond \equiv ?(|\mathsf{x}| < k-\varepsilon); \mathsf{x}{:=}\mathsf{x}{+}h\mathsf{v}; \varepsilon{:=}(1{+}h\ell)\varepsilon{+}\frac{\ell m}{2}h^2$ describes a single iteration step of an Euler approximation with step size h and accumulates the standard local error bound in the variable ε. The program ensures that the approximation remains in the compact domain $|\mathsf{x}| < k - \varepsilon$.

Theorem 4 (Euler Axiom Soundness). *The Euler axiom $\langle E\rangle$ is sound*

$$\langle E\rangle \quad \begin{array}{l}\langle \mathsf{x}' = \mathsf{v}\,\&\,\rho^\diamond\rangle\varphi^\diamond \\ \leftrightarrow \exists k,m,\ell{>}0\,(\text{bound}^\diamond \wedge \exists h{>}0\,\langle\varepsilon{:=}0; (?\rho^\diamond_{\mathsf{x}\pm\varepsilon}; \text{step}^\diamond)^*\rangle(\varphi^\diamond_{\mathsf{x}\pm\varepsilon} \wedge \varepsilon < 1))\end{array}$$

where $\varphi^\diamond$ is a strict $\mathsf{dL}_{\mathsf{r}\diamond}$ formula and $h,k,m,\ell,\varepsilon \notin FV(\varphi^\diamond) \cup \{\mathsf{x},\mathsf{v}\}$ are fresh.

See proof on page 363.

Observe that in $\langle E\rangle$ both sides of the equivalence are strict formulas. It is critical for soundness that the postcondition $\varphi^\diamond$ is syntactically a strict formula and, thus, semantically an *open set*. This allows some (arbitrarily small) wiggle room and sound, symbolic approximations become feasible, since the robust postcondition $\varphi^\diamond$ can be equivalently replaced by a quantitatively robust strengthening $\varphi^\diamond_{\mathsf{x}\pm\varepsilon}$. For soundness of the forward direction, it suffices to show that the error accumulation correctly bounds the error. For the converse, the evolution bound k plays a critical role in ensuring solution existence. Within the bounds, the Euler approximation remains close to the real solution in the region $|\mathsf{x}| < k+2$ and does not leave before reaching $\varphi^\diamond_{\mathsf{x}\pm\varepsilon}$ by the iterative assumption. Hence, the actual solution must remain bounded before reaching $\varphi^\diamond_{\mathsf{x}\pm\varepsilon}$. In particular, the solution exists at least until it reaches $\varphi^\diamond$. The Euler axiom $\langle E\rangle$ is closely related to the differential equation characterization axiom $\overleftrightarrow{\Delta}$ [8], which does not handle the possibility of divergent evolutions and postconditions with modalities.

To summarize, the proof calculus for dL and, thus, for $\mathsf{dL}_{\mathsf{r}\diamond}$ is based on the basic rule $\mathbb{R}$, which discharges valid formulas of real arithmetic. The calculus

extends $\mathbb{R}$ and basic propositional logical rules by axioms for (discrete) hybrid programs and axioms for continuous evolution including $[\text{ODE}\rangle$ and $\langle E\rangle$.

4 Completeness for Robust Hybrid Systems Reachability

The main result of this paper is the proof of completeness of robust reachability differential dynamic logic. Unlike previous completeness proofs, it does not rely on an undecidable oracle or apply only to algebraic hybrid systems [8,12]. Completeness for $\mathsf{dL}_{\mathsf{r}\diamond}$ is proved by an inductive argument, which step-by-step reduces the complexity of a formula to a basic formula of real arithmetic, which can be discharged by rule $\mathbb{R}$ and decided computationally [2]. There are two sources of difficulty. First, the induction is transfinite, as it needs to handle all finite iterations α^n before handling loop programs α^*. Second, the topological robustness properties need to be exploited to reduce all formulas to simpler valid ones. The loops pose a particular challenge, as they cannot be reduced immediately to *semantically equivalent* simpler formulas.

The proof of the following completeness theorem is by transfinite induction on a well-founded order defined in Section 4.1. The proof is sketched in Section 4.2, and the complete proof is in Section D.

Theorem 5 (Completeness for Robust Reachability Formulas). *Every valid strict robust reachability sentence $\varphi^\diamond$ is provable: $\vDash \varphi^\diamond$ iff $\vdash \varphi^\diamond$.*

4.1 Induction Order

The transfinite induction is based on an order on all dL formulas, which is used to sequence the induction steps. This order makes it possible to reduce all formulas to simpler ones in a way that is compatible with the proof calculus. To define the order, every formula and every program of dL is assigned an *ordinal rank* $\operatorname{rank}(\cdot)$ by structural induction and the relations $\preccurlyeq$ and $\prec$ are defined by

$$\varphi \prec \rho \iff \operatorname{rank}(\varphi) < \operatorname{rank}(\rho) \qquad \varphi \preccurlyeq \rho \iff \operatorname{rank}(\varphi) \leq \operatorname{rank}(\rho)$$

The full formal definition of the rank is in Section C. The important properties of the order for the inductive proof are summarized in Proposition 6.

Proposition 6. *The relation $\prec$ on the set of dL formulas is well-founded and*

1. $\varphi, \rho \preccurlyeq \varphi \vee \rho$ *and* $\varphi, \rho \prec \varphi \wedge \rho$
2. $\langle x{:=}v\rangle\varphi \prec \exists x\, \varphi$ *and* $\varphi \prec \forall x\, \varphi$ *for all terms* v
3. $\varphi\frac{v}{x} \prec \langle x{:=}v\rangle\varphi$
4. $\rho \wedge \varphi \prec \langle ?\rho\rangle\varphi$
5. $\langle\alpha\rangle\varphi \vee \langle\gamma\rangle\varphi \prec \langle\alpha \cup \gamma\rangle\varphi$
6. $\langle\alpha\rangle\langle\gamma\rangle\varphi \prec \langle\alpha;\gamma\rangle\varphi$
7. $\langle\alpha^{\leq n}\rangle\varphi \prec \langle\alpha^*\rangle\varphi$
8. $\langle(?\rho_{\mathsf{x}\pm\varepsilon};\mathsf{step}^\diamond)^*\rangle(\varphi_{\mathsf{x}\pm\varepsilon} \wedge \varepsilon < 1) \prec \langle \mathsf{x}' = \mathsf{v}\,\&\,\rho \wedge |\mathsf{x}| < k\rangle\varphi$

As the completeness proof proceeds by induction on sequents, the order is extended to sets of formulas. Recall that both sides of a sequent $\Delta \vdash \Gamma$ consist of a finite (unordered) sets of dL formulas Γ, Δ. On the set Σ of all sets of finite (unordered) sets of formulas define the *induction order* $\prec$ and $\preccurlyeq$ as follows:

$$\Gamma_1 \prec \Gamma_2 \iff \max\{\mathrm{rank}(\varphi) : \varphi \in \Gamma_1\} < \max\{\mathrm{rank}(\varphi) : \varphi \in \Gamma_2\}$$
$$\Gamma_1 \preccurlyeq \Gamma_2 \iff \max\{\mathrm{rank}(\varphi) : \varphi \in \Gamma_1\} \leq \max\{\mathrm{rank}(\varphi) : \varphi \in \Gamma_2\}$$

The order $\prec$ is well-founded on Σ. If $\Gamma_1 \prec \Gamma_3$ and $\Gamma_2 \prec \Gamma_3$, then $\Gamma_1, \Gamma_2 \prec \Gamma_3$ and the order is invariant under variable substitutions.

4.2 Proof of Completeness

Finally, the induction order can be used to prove completeness for $\mathsf{dL}_{\mathsf{r}\diamond}$ step-by-step. This section provides some intuition and the full proof can be found in Section D. While the completeness result is stated only for sentences, the inductive proof shows more generally that every *valid* sequent of the form

$$x_1 \in I_1, \ldots, x_n \in I_n \vdash \Gamma \quad \text{for } \mathrm{FV}(\Gamma) \subseteq \{x_1, \ldots, x_n\},\ \Gamma \text{ strict } \mathsf{dL}_{\mathsf{r}\diamond} \text{ formulas}$$

is provable. The assumptions on the left-hand side bound all the free variables of Γ. The proof proceeds by induction on Γ with respect to the $\prec$ order. In the inductive step, the strict $\mathsf{dL}_{\mathsf{r}\diamond}$ formulas in Γ are shown to be derivable from valid sequents with $\prec$-lower right-hand side. The topological interpretation of robustness (Theorem 2) plays an important role. For example, when reducing an existential quantifier in a valid sentence $\exists x\, \varphi^\circ$, there is a *rational witness* q, since φ° denotes an open set (Theorem 2). As rationals are expressible syntactically, the rule $\exists$R can be used to instantiate the quantifier. This retains validity and reduces the rank of the formula by Proposition 6(2).

The most interesting and crucial step is the case of loop programs α^* in diamond modalities. (Loops in box modalities are not allowed in $\mathsf{dL}_{\mathsf{r}\diamond}$.) Since all free variables are bounded by the left-hand side of the sequent, the number of iterations needed can be bounded by a *constant* using a compactness argument. In this way, an unbounded loop can be reduced via $\langle{\leq n}\rangle$ to a bounded loop, which is of lower complexity by Proposition 6(7).

A continuous program in a diamond modality can be reduced to the case of an iteration program by axiom $\langle E \rangle$. Since continuous programs in box modalities have bounded evolution domains, the case of such programs reduces to the case of continuous programs in diamond modalities by axiom [ODE⟩.

An interesting corollary to completeness is that the validity of $\mathsf{dL}_{\mathsf{r}\diamond}$ sentences and, thus, of robust dL reachability problems of hybrid systems is semi-decidable:

Corollary 7. *Validity of strict* $\mathsf{dL}_{\mathsf{r}\diamond}$ *sentences* φ° *is semi-decidable.*

In fact, the completeness proof gives a constructive algorithm for constructing a witnessing proof tree computably and proof search terminates. If a strict $\mathsf{dL}_{\mathsf{r}\diamond}$ sentence φ° is robustly determined and *all* quantifiers are bounded, then one of the strict sentences φ° or $\sim\varphi^\circ$ is valid:

Corollary 8. *Validity of robustly determined, bounded strict* $\mathsf{dL}_{\mathsf{r}\diamond}$ *sentences is decidable.*

However, in general, it is difficult to know whether a given strict $\mathsf{dL}_{\mathsf{r}\diamond}$ sentence is robustly determined or not.

The *doubly exponential* computational complexity of quantifier elimination algorithms [3], which underlie the proof rule $\mathbb{R}$, is one of the major bottlenecks for scalable symbolic hybrid systems verification. However, the completeness (Theorem 5) of $\mathsf{dL}_{\mathsf{r}\diamond}$ relies only on the rule $\mathbb{R}$ for the computationally more tractable case of *quantifier-free strict* formulas, which has *singly exponential* complexity [14].

5 Conclusion

The paper bridges the long-standing gap between approximate approaches (e.g., δ-decidability) and exact deductive approaches to hybrid systems reachability verification. By identifying robust differential dynamic logic as a syntactic fragment of dL, the exact semantics is retained and *absolute* completeness is gained. Unlike in previous completeness proofs, *no undecidable oracle* is required. The central insight is that robustness is ensured by a simple syntactic restriction in dL that enforces strict inequalities. Crucially, robustness is purely *topological* and does not involve quantitative noise tolerance and margins of error. Consequently, working with robust dL formulas entails no loss of convenience or practical expressiveness compared to standard dL.

This work opens several areas of future research. In particular, further investigation of the robust fragment for *safety* questions is promising. Moreover, an implementation of a proof tactic in the KeYmaera X theorem prover for dL can be an impactful tool to automatically verify robust reachability properties.

Acknowledgments. This work has been supported by an Alexander von Humboldt Professorship. We thank the anonymous reviewers for their detailed comments, which helped to improve the paper.

References

1. Cassez, F., Larsen, K.G.: The impressive power of stopwatches. In: CONCUR. pp. 138–152 (2000). https://doi.org/10.1007/3-540-44618-4_12
2. Collins, G.E.: Quantifier elimination for real closed fields by cylindrical algebraic decompostion. In: Brakhage, H. (ed.) Automata Theory and Formal Languages. pp. 134–183. Springer, Berlin, Heidelberg (1975)
3. Davenport, J.H., Heintz, J.: Real quantifier elimination is doubly exponential. J. Symb. Comput. **5**(1/2), 29–35 (1988). https://doi.org/10.1016/S0747-7171(88)80004-X
4. Fränzle, M.: Analysis of hybrid systems: An ounce of realism can save an infinity of states. In: Flum, J., Rodríguez-Artalejo, M. (eds.) Computer Science Logic, 13th International Workshop, CSL '99, 8th Annual Conference of the EACSL, Madrid,

Spain, September 20-25, 1999, Proceedings. LNCS, vol. 1683, pp. 126–140. Springer (1999). https://doi.org/10.1007/3-540-48168-0_10
5. Henzinger, T.A., Kopke, P.W., Puri, A., Varaiya, P.: What's decidable about hybrid automata? In: Leighton, F.T., Borodin, A. (eds.) Proceedings of the Twenty-Seventh Annual ACM Symposium on Theory of Computing, 29 May-1 June 1995, Las Vegas, Nevada, USA. pp. 373–382. ACM (1995). https://doi.org/10.1145/225058.225162
6. Kong, S., Gao, S., Chen, W., Clarke, E.M.: dReach: δ-reachability analysis for hybrid systems. In: Baier, C., Tinelli, C. (eds.) Tools and Algorithms for the Construction and Analysis of Systems, TACAS 2015. LNCS, vol. 9035, pp. 200–205. Springer (2015). https://doi.org/10.1007/978-3-662-46681-0_15
7. Platzer, A.: Differential dynamic logic for hybrid systems. J. Autom. Reason. **41**(2), 143–189 (2008). https://doi.org/10.1007/S10817-008-9103-8
8. Platzer, A.: The complete proof theory of hybrid systems. In: Proceedings of the 27th Annual IEEE Symposium on Logic in Computer Science, LICS 2012, Dubrovnik, Croatia, June 25-28, 2012. pp. 541–550. IEEE Computer Society (2012). https://doi.org/10.1109/LICS.2012.64
9. Platzer, A.: A uniform substitution calculus for differential dynamic logic. In: Felty, A.P., Middeldorp, A. (eds.) Automated Deduction - CADE-25 - 25th International Conference on Automated Deduction, Berlin, Germany, August 1-7, 2015, Proceedings. LNCS, vol. 9195, pp. 467–481. Springer (2015). https://doi.org/10.1007/978-3-319-21401-6_32
10. Platzer, A.: A complete uniform substitution calculus for differential dynamic logic. J. Autom. Reason. **59**(2), 219–265 (2017). https://doi.org/10.1007/S10817-016-9385-1
11. Platzer, A., Qian, L.: Axiomatization of compact initial value problems: Open properties. J. ACM **72**(6), 41:1–41:51 (2025). https://doi.org/10.1145/3763228
12. Platzer, A., Tan, Y.K.: Differential equation invariance axiomatization. J. ACM **67**(1), 6:1–6:66 (2020). https://doi.org/10.1145/3380825
13. Ratschan, S.: Safety verification of non-linear hybrid systems is quasi-decidable. Formal Methods Syst. Des. **44**(1), 71–90 (2014). https://doi.org/10.1007/S10703-013-0196-2
14. Renegar, J.: On the computational complexity and geometry of the first-order theory of the reals, part I: introduction. preliminaries. the geometry of semi-algebraic sets. the decision problem for the existential theory of the reals. J. Symb. Comput. **13**(3), 255–300 (1992). https://doi.org/10.1016/S0747-7171(10)80003-3
15. Schmitt, P.H.: Diamond formulas: A fragment of dynamic logic with recursively enumerable validity problem. Inf. Control. **61**(2), 147–158 (1984). https://doi.org/10.1016/S0019-9958(84)80055-8
16. Tan, Y.K., Platzer, A.: Switched systems as hybrid programs. In: Jungers, R.M., Ozay, N., Abate, A. (eds.) 7th IFAC Conference on Analysis and Design of Hybrid Systems, ADHS 2021, Brussels, Belgium, July 7-9, 2021. pp. 247–252. No. 5 in IFAC-PapersOnLine, Elsevier (2021). https://doi.org/10.1016/J.IFACOL.2021.08.506
17. Tarski, A.: A decision method for elementary algebra and geometry. In: Caviness, B.F., Johnson, J.R. (eds.) Quantifier Elimination and Cylindrical Algebraic Decomposition. pp. 24–84. Springer, Vienna (1998)
18. Walter, W.: Ordinary Differential Equations, Graduate Texts in Mathematics, vol. 182. Springer, New York (1998). https://doi.org/10.1007/978-1-4612-0601-9

19. Weihrauch, K.: Computable Analysis - An Introduction. Texts in Theoretical Computer Science. An EATCS Series, Springer (2000). https://doi.org/10.1007/978-3-642-56999-9

A Polarity

The negation of a robust property is not necessarily itself robust. For this reason the (syntactic) polarity of a subformula is important when defining which properties are robust. An instance of a subformula φ inside a formula ρ is said to appear with *positive polarity* if it is within the scope of an even number of negations and box modalities. (Since box modalities and tests can be nested.) Otherwise, it is said to appear with *negative polarity*. To understand why box modalities affect polarity, consider, for example, the formula $[?\varphi]\rho$. As this formula is equivalent to $\neg\varphi \vee \rho$, it is important that the subformula φ is considered to appear in negative polarity.

B Evolution Domain Axioms

Evolution Domain in Time. Although the completeness proof handles the *evolution domains* φ of a continuous program $\mathsf{x}' = \mathsf{v} \,\&\, \varphi$ directly, it is illuminating to see that it can be reduced to an evolution domain free diamond property and a *time-bounded* box property.

Theorem 9 (Soundness &). *The evolution domain axiom* $\langle\&\rangle$ *is sound:*

$$\langle\&\rangle \quad \langle \mathsf{x}' = \mathsf{v} \,\&\, \rho\rangle\varphi \leftrightarrow \exists \tau_0 \, (\langle \mathsf{x}' = \mathsf{v}\rangle(\varphi \wedge \tau = \tau_0) \wedge [\mathsf{x}' = \mathsf{v} \,\&\, \tau \leq \tau_0]\rho)$$

where τ_0 *does not appear freely in* $x, FV(\mathsf{v}), FV(\varphi)$ *and* $FV(\rho)$.

Proof. As in the definition of the semantics for continuous programs let $f_\omega : \mathbb{R}^n \to \mathbb{R}^n$ be the vector field defined by $f_\omega(\mathsf{a}) = \omega^{\mathsf{a}}_{\mathsf{x}}[\![\mathsf{v}]\!]\restriction\mathsf{x}$ and define the state-flow map $\Psi_\omega(t) = \omega^{\Phi_{f_\omega}(\omega\restriction\mathsf{x},t)}_{\mathsf{x}}$.

To prove soundness of the forward implication, consider $\omega \in [\![\langle \mathsf{x}' = \mathsf{v} \,\&\, \rho\rangle\varphi]\!]$. Then by definition there is t such that $(\omega, t) \in \Omega_{f_\omega}$, $\Psi_\omega([0,t]) \subseteq [\![\rho]\!]$ and $\Psi_\omega(t) \in [\![\varphi]\!]$. Then $\nu = \omega^{\omega[\![\tau]\!]+t}_{\tau_0}$ satisfies $\nu \in [\![\langle \mathsf{x}' = \mathsf{v}\rangle(\varphi \wedge \tau = \tau_0) \wedge [\mathsf{x}' = \mathsf{v} \,\&\, \tau \leq \tau_0]\rho]\!]$.

Conversely, consider ω such that there is t with $\nu = \omega^{\omega[\![\tau]\!]+t}_{\tau_0}$ satisfying $\nu \in [\![\langle \mathsf{x}' = \mathsf{v}\rangle(\varphi \wedge \tau = \tau_0) \wedge [\mathsf{x}' = \mathsf{v} \,\&\, \tau \leq \tau_0]\rho]\!]$. Then there is s such that $\Psi_\nu(s) \in [\![\varphi]\!] \cap [\![\tau = \tau_0]\!]$. Thus $t = s$ and since $\nu \in [\![[\mathsf{x}' = \mathsf{v} \,\&\, \tau \leq \tau_0]\rho]\!]$ also $\Psi_\nu([0,t]) \in [\![\rho]\!]$. So ν belongs to $[\![\langle \mathsf{x}' = \mathsf{v} \,\&\, \rho\rangle\varphi]\!]$ and so does ω as τ_0 is not free in the formula. □

Differential Equation Evolution Bound Axiom. It is often useful to assume boundedness of the evolution domain constraint. For reachability properties a suitable bound always exists and can be introduced syntactically with axiom $|\&|$.

Proposition 10 (Evolution Domain Bound). *The axiom* |&| *is sound:*

$$|\&| \quad \langle \mathsf{x}' = \mathsf{v} \,\&\, \rho\rangle\varphi \leftrightarrow \exists y\, \langle \mathsf{x}' = \mathsf{v} \,\&\, \rho \wedge |\mathsf{x}| < y\rangle\varphi \qquad (y \notin \varphi, \rho, \mathsf{x}, \mathsf{v})$$

Proof. For the forward direction consider $\omega \in [\![\langle \mathsf{x}' = \mathsf{v} \,\&\, \rho\rangle\varphi]\!]$ and define $f_\omega(\mathsf{a}) = \omega^{\mathsf{a}}_{\mathsf{x}}[\![\mathsf{v}]\!]\restriction\mathsf{x}$ and $\Psi_\omega(t) = \omega^{\Phi_{f_\omega}(\omega\restriction\mathsf{x},t)}_{\mathsf{x}}$ as in the definition of the semantics of continuous programs. There is some $t \geq 0$ such that $\Psi_\omega(t) \in [\![\varphi]\!]$ and $\Psi_\omega([0,t]) \in [\![\rho]\!]$. By compactness of $\Psi_\omega([0,t])$ there is some $K > \|\Psi_\omega(s)\|_\infty$ for all $s \in [0,t]$. Hence $\omega^K_y \in [\![\langle \mathsf{x}' = \mathsf{v} \,\&\, \rho \wedge |\mathsf{x}| < y\rangle\varphi]\!]$.

Soundness of the backward implication $\leftarrow$ is immediate. □

Note that both sides of axiom |&| are strict formulas if φ, ρ are strict.

C Definition of Rank Function

To define the order of induction, assign an ordinal rank to every formula and every program by mutual induction on formulas and programs.

$$\begin{aligned}
&\mathrm{rank}(\varphi \vee \rho) = \mathrm{rank}(\varphi \wedge \rho) = \max\{\mathrm{rank}(\varphi), \mathrm{rank}(\rho)\} + 1\\
&\mathrm{rank}(\exists x\, \varphi) = \mathrm{rank}(\varphi) + 5 \qquad \mathrm{rank}(\forall x\, \varphi) = \mathrm{rank}(\varphi) + 1\\
&\mathrm{rank}(\langle\alpha\rangle\varphi) = \mathrm{rank}([\alpha]\varphi) = \mathrm{rank}(\varphi) + \mathrm{rank}(\alpha) + 1
\end{aligned}$$

Note that since ordinal addition is not commutative, the order of addition in $\mathrm{rank}(\langle\alpha\rangle\varphi) = \mathrm{rank}(\varphi) + \mathrm{rank}(\alpha) + 1$ is important. For programs:

$$\begin{aligned}
&\mathrm{rank}(x{:=}v) = 4\\
&\mathrm{rank}(\alpha \cup \gamma) = \mathrm{rank}(\alpha) + \mathrm{rank}(\gamma) + 2\\
&\mathrm{rank}(?\varphi) = \mathrm{rank}(\varphi) + 1\\
&\mathrm{rank}(\alpha;\gamma) = \mathrm{rank}(\gamma) + 1 + \mathrm{rank}(\alpha) + 1\\
&\mathrm{rank}(\mathsf{x}' = \mathsf{v} \,\&\, \varphi) = (\mathrm{rank}(\varphi) + 17) \cdot \omega\\
&\mathrm{rank}(\alpha^*) = (\mathrm{rank}(\alpha) + 1) \cdot \omega + 1
\end{aligned}$$

where ω is the first limit ordinal.

D Proof of Completeness for Robust Reachability

Proof of Theorem 5. First, by repeatedly applying axioms $\langle{:=}\rangle$, $\langle ? \rangle$, $\langle\cup\rangle$ and $\langle;\rangle$ in context any $\mathsf{dL}_{\mathsf{r}\diamond}$ sentence is provably equivalent to a sentence with box modalities only in the form $[\mathsf{x}' = \mathsf{v} \,\&\, \rho^\bullet \wedge |\mathsf{x}| \leq k]\varphi^\circ$. Now applying the differential equation duality axiom [ODE⟩ in context, any $\mathsf{dL}_{\mathsf{r}\diamond}$ sentence is provably equivalent to an $\mathsf{dL}_{\mathsf{r}\diamond}$ sentence without box modalities.

The proof proceeds by induction on a particular class of sequents potentially mentioning free variables. A sequent is *robust* if it is of the form $\Delta \vdash \Gamma$, where Γ consists of box-free strict formulas and Δ contains a formula of the form $(v \leq x \wedge x \leq w) \in \Delta$ for every $x \in \mathrm{FV}(\Gamma)$ where v, w are terms not mentioning x. (The formula $x \in I_x$ abbreviates $v \leq x \wedge x \leq w$.) The inductive step is encapsulated in the following lemma:

Lemma 11. *If $\Delta \vdash \Gamma$ is a valid robust sequent, then there are valid robust sequents $\Delta_1 \vdash \Gamma_1, \dots, \Delta_n \vdash \Gamma_n$ such that $\Gamma_1 \prec \Gamma, \dots, \Gamma_n \prec \Gamma$ and*

$$\frac{\Delta_1 \vdash \Gamma_1 \quad \dots \quad \Delta_n \vdash \Gamma_n}{\Delta \vdash \Gamma}$$

is derivable in the dL *calculus.*

The proof of this lemma is by another sub-induction. First, a related claim only about the potentially reducible non-basic formulas is shown. A formula is *basic* if it does not contain any modalities, i.e., if it is a formula of real arithmetic. A set Γ of formulas is non-basic if it does not contain a basic formula.

Lemma 12. *Let $\Delta \vdash \Gamma, \Gamma'$ be a valid robust sequent and Γ is non-basic. There are valid robust sequents $\Delta_1 \vdash \Gamma_1, \Gamma', \dots, \Delta_n \vdash \Gamma_n, \Gamma'$ with $\Gamma_1 \prec \Gamma, \dots, \Gamma_1 \prec \Gamma$ and* dL *derives:*

$$\frac{\Delta_1 \vdash \Gamma_1, \Gamma' \quad \dots \quad \Delta_n \vdash \Gamma_n, \Gamma'}{\Delta \vdash \Gamma, \Gamma'}$$

Proof of Lemma 12. This claim is shown by induction on the length of Γ for all Γ'. So suppose the claim holds for Γ and consider a valid robust sequent $\Delta \vdash \varphi^\diamond, \Gamma, \Gamma'$. By the induction hypothesis on Γ, it suffices to find a list of valid robust sequents $\Delta_1 \vdash \Gamma_1, \Gamma, \Gamma', \dots, \Delta_n \vdash \Gamma_n, \Gamma, \Gamma'$ with $\Gamma_1 \prec \varphi^\diamond, \dots, \Gamma_1 \prec \varphi^\diamond$ such that dL derives the following

$$\frac{\Delta_1 \vdash \Gamma_1, \Gamma, \Gamma' \quad \dots \quad \Delta_n \vdash \Gamma_n, \Gamma, \Gamma'}{\Delta \vdash \varphi^\diamond, \Gamma, \Gamma'}$$

To find such sequents proceed by another induction on the length of formula φ.
- *Case* $\varphi_1^\diamond \vee \varphi_2^\diamond$: By applying the claim to the formulas $\varphi_1^\diamond$ and $\varphi_2^\diamond$, which are shorter, note that dL derives

$$\vee\text{R}\,\frac{\dfrac{\dfrac{\Delta_1 \vdash \Gamma_1^1, \Gamma_1^2, \Gamma, \Gamma' \quad \dots}{\Delta_1 \vdash \Gamma_1^1, \varphi_2^\diamond, \Gamma, \Gamma'} \quad \dots}{\Delta \vdash \varphi_1^\diamond, \varphi_2^\diamond, \Gamma, \Gamma'}}{\Delta \vdash \varphi_1^\diamond \vee \varphi_2^\diamond, \Gamma, \Gamma'}$$

Validity is preserved by the inductive hypothesis and $\Gamma_i^1 \prec \varphi_1^\diamond$ and $\Gamma_i^2 \prec \varphi_2^\diamond$ so $\Gamma_i^1, \Gamma_i^2 \prec \varphi_1^\diamond, \varphi_2^\diamond \preccurlyeq \varphi_1^\diamond \vee \varphi_2^\diamond$ by Proposition 6(1).
- *Case* $\varphi_1^\diamond \wedge \varphi_2^\diamond$: Letting $\Delta_1 = \Delta$ and $\Gamma_1 = \{\varphi_1^\diamond\}$ and $\Gamma_2 = \{\varphi_2^\diamond\}$ is sufficient by $\wedge$R and Proposition 6(1) since both sequents are clearly valid.
- *Case* $\exists x\, \varphi^\diamond$: Note that as $[\![\varphi^\diamond]\!]$ is open and the sequent is valid, there is a rational witness $q \in \mathbb{Q}$ for x. Choosing $\Delta_1 = \Delta$ and $\Gamma_1 = \{\varphi^\diamond{}_x^q\}$ is as required, by $\exists$R and Proposition 6(2).
- *Case* $\forall_I x\ \varphi^\diamond$: By uniform renaming, without loss of generality, x does not appear in Δ_1, Γ or Γ'. So letting $\Delta_1 = \Gamma \cup \{x \in I\}$ and $\Gamma_1 = \{\varphi^\diamond\}$ works by $\forall$R and Proposition 6(2).

- *Case* $\langle\alpha^\diamond\rangle$: Distinguish based on the top level connective of $\alpha^\diamond$. For the cases of assignment $x{:=}v$, test $?\varphi^\diamond$, choice $\alpha_1^\diamond \cup \alpha^\diamond$ and composition programs $\alpha_1^\diamond;\alpha^\diamond$ the reduction is straightforward by axioms $\langle:=\rangle$, $\langle?\rangle$, $\langle\cup\rangle$, $\langle;\rangle$ and properties (3), (4), (5), (6) of Proposition 6, respectively. Only the cases of infinite loops and differential equations remain.
- *Case* $\langle\alpha^*\rangle\varphi^\diamond$: Note that validity of sequent $\Delta \vdash \langle\alpha^{\diamond *}\rangle\varphi, \Gamma, \Gamma'$ means that $\bigcap_{\rho^\bullet\in\Delta}[\![\rho^\bullet]\!] \cap \bigcap_{\rho^\diamond\in\Gamma\cup\Gamma'}[\![\rho^\diamond]\!]^{\mathrm{c}} \subseteq \bigcup_{n=0}^{\infty}[\![\langle(\alpha^\diamond)^{\leq n}\rangle\varphi]\!]$. Note that the left hand side of this inclusion is closed as the intersection of closed sets by Theorem 2. Since Δ bounds all relevant variables, we can assume without loss of generality that the left hand side is compact. Since each $[\![\langle(\alpha^\diamond)^{\leq n}\rangle\varphi]\!]$ is open, there is some n such that $\bigcap_{\rho^\bullet\in\Delta}[\![\rho^\bullet]\!] \cap \bigcap_{\rho^\diamond\in\Gamma\cup\Gamma'}[\![\rho^\diamond]\!]^{\mathrm{c}} \subseteq [\![\langle(\alpha^\diamond)^{\leq n}\rangle\varphi]\!]$. Hence $\Delta \vdash \langle(\alpha^\diamond)^{\leq n}\rangle\varphi, \Gamma, \Gamma'$ is valid and choosing $\Delta_1 = \Delta$ and $\Gamma_1 = \{\langle(\alpha^\diamond)^{\leq n}\rangle\varphi^\diamond\}$ is as required by $\langle^{\leq n}\rangle$ and Proposition 6(7).
- *Case* $\langle \mathsf{x}' = \mathsf{v}\,\&\,\varphi^\diamond\rangle$: By using the Euler axiom $\langle E\rangle$ and reducing the fresh logical connectives as in the previous cases, the sequent can be reduced to one of the form

$$|\&|\ \frac{\dfrac{\Delta, \varepsilon = 0 \vdash \langle(?\rho^\diamond_{\mathsf{x}\pm\varepsilon};\mathrm{step}^\diamond)^*\rangle(\varphi^\diamond_{\mathsf{x}\pm\varepsilon} \wedge \varepsilon < 1), \Gamma, \Gamma'}{\vdots}}{\Delta \vdash \langle \mathsf{x}' = \mathsf{v}\,\&\,\rho^\diamond\rangle\varphi^\diamond, \Gamma, \Gamma'}$$

where k, m, ℓ and h have been replaced by rational constants. The claim follows by Proposition 6(8). □

Proof of Lemma 11. Consider a valid robust sequent $\Delta \vdash \Gamma$ and let Γ_0 be the non-basic and Γ' be the basic formulas of Γ. If $\Gamma_0 = \emptyset$, then $\Delta \vdash \Gamma$ is derivable by $\mathbb{R}$ as it is a valid sequent of first-order real arithmetic. In other words the empty list of robust sequents derives $\Delta \vdash \Gamma$ and witnesses the lemma.

If $\Gamma_0 \neq \emptyset$, by Lemma 12 there are valid robust sequents $\Delta_1 \vdash \Gamma_1, \Gamma', \ldots, \Delta_n \vdash \Gamma_n, \Gamma'$ from which dL derives $\Delta \vdash \Gamma_0, \Gamma'$ and $\Gamma_1 \prec \Gamma_0, \ldots, \Gamma_1 \prec \Gamma_0$. As the rank of the basic formulas is 0, moreover $\Gamma_1, \Gamma' \prec \Gamma$ and $\ldots \Gamma_n, \Gamma' \prec \Gamma$. □

To conclude the proof of Theorem 5, apply Lemma 11 repeatedly, beginning with the valid robust sequent $\vdash \varphi$ for a box-free strict $\mathsf{dL}_{\mathsf{r}\diamond}$ sentence. The result is a (possibly infinite) derivation tree for $\vdash \varphi$. If this tree were infinite, there would be an infinite branch through the constructed tree by König's Lemma. However, every application of Lemma 11 on this branch reduces the succedent of the sequent with respect to $\prec$ by the construction (Lemma 11). Such an infinite $\prec$-descending sequence in Σ contradicts well-foundedness of $\prec$. So the derivation is finite and as such the required proof. □

E Additional Proofs

Proof of Lemma 1. Fix some $x \in X$. For every $y \in Y$ pick open neighborhoods V_y, W_y such that $(x, y) \in V_y \times W_y \subseteq U$. As Y is compact, there are $y_1, \ldots, y_n$ such that $Y \subseteq W_{y_1} \cup \ldots \cup W_{y_n}$. Let $V = V_{y_1} \cap \ldots \cap V_{y_n}$ and note that V is open, $x \in V$ and $(x, y) \in U$ for all $y \in Y$. □

Proof of Theorem 2. Proceed by induction on the formulas and programs of $\mathsf{dL}_{\mathsf{r}\diamond}$. Many cases are immediate, and only the interesting ones are shown.
- *Case* (1): For a formula $\exists x\, \varphi^\diamond$ note that $[\![\exists x\, \varphi^\diamond]\!] = \bigcup_{a\in\mathbb{R}} \{\omega : \omega_x^a \in [\![\varphi^\diamond]\!]\}$ is open as a union of open sets.

Most interesting is the case for a bounded universal quantifier $\forall_I x\, \varphi^\diamond$ where I is a closed (syntactic) interval with terms $[v, w]$. For every state ω define the continuous function $\gamma(s) = s\omega[\![v]\!] + (1-s)\omega[\![w]\!]$, so that

$$[\![\forall_I x\, \varphi^\diamond]\!] = \{\omega : \forall s{\in}[0,1]\ \omega_x^{\gamma(s)} \in [\![\varphi^\diamond]\!]\}.$$

This set is open by Lemma 1, since $\{(\omega, s) \in \mathcal{S} \times [0,1] : \omega_x^{\gamma(s)} \in [\![\varphi^\diamond]\!]\}$ is open with respect to the topology on $\mathcal{S} \times [0,1]$.

For diamond and box formulas openness follows from the induction hypotheses (3) and (4).
- *Case* (2): This is the dual of the previous case.
- *Case* (3): For $\alpha^{\diamond *}$ note that $[\![\alpha^{\diamond *}]\!]^{-1}(U) = \bigcup_{n\geq 0} [\![\alpha^\diamond]\!]^n(U)$ is open as a union of open sets.

For a continuous program $\mathsf{x}' = \mathsf{v}\,\&\,\varphi^\diamond$, as in the definition of the semantics, let $f_\omega : \mathbb{R}^n \to \mathbb{R}^n$ be defined by $f_\omega(\mathsf{a}) = \omega_{\mathsf{x}}^{\mathsf{a}}[\![\mathsf{v}]\!]\restriction_{\mathsf{x}}$ and set $\Psi_\omega(t) = \omega_{\mathsf{x}}^{\Phi_{f_\omega}(\omega\restriction_{\mathsf{x}}, t)}$. For continuous programs define the set of states reaching U in time t as

$$R_t(U) = \{\omega : \Psi_\omega(t) \in U\} \cap \{\omega : \forall s{\in}[0,t]\ \Psi_\omega(s) \in [\![\varphi^\diamond]\!]\}$$

so that $[\![\mathsf{x}' = \mathsf{v}\,\&\,\varphi^\diamond]\!]^{-1}(U) = \bigcup_{t\geq 0} R_t(U)$. Hence, it suffices to show that every $R_t(U)$ is open. As the flow Φ_{f_ω} is continuous in ω, so is Ψ_ω, and hence $\{\omega : \Psi_\omega(t) \in U\}$ and $\{(\omega, t) : \Psi_\omega(t) \in [\![\varphi^\diamond]\!]\}$ are open, since U and $[\![\varphi^\diamond]\!]$ are open by assumption and induction hypothesis, respectively. Moreover, by Lemma 1 the set $\{\omega : \forall s{\in}[0,t]\ \Phi_x(\omega, s) \in [\![\varphi^\diamond]\!]\}$ is open.
- *Case* (4): The only case of interest is for continuous programs $\mathsf{x}' = \mathsf{v}\,\&\,\varphi^\bullet \wedge |\mathsf{x}| \leq k$. Define the functions f_ω and Ψ_ω as before. Since k does not mention any variables from x the evolution, $K = \omega[\![k]\!]$ is constant, i.e., $\Psi_\omega[\![t]\!](k) = K$ for all t. Similar to the previous case, for a closed set C define

$$R_t(C) = \{\omega : \Psi_\omega(t) \in C\} \cap \{\omega : \forall s \in [0,t]\ \Psi_\omega(s) \in [\![\varphi^\bullet]\!]\,,\ \|\Psi_\omega(s)\|_\infty \leq K\}$$

and again every $R_t(C)$ is closed by continuity of Ψ_ω. Since the differential equation contains $\tau' = 1$, the potential evolutions are time-bounded by K, so that $[\![\mathsf{x}' = \mathsf{v}\,\&\,\varphi^\bullet \wedge |\mathsf{x}| \leq k]\!]^{-1}(C) = \{\omega : \exists t{\in}[0,K]\ \omega \in R_t(C)\}$. By Lemma 1 this set is closed. □

Proof of Theorem 3. As in the definition of the semantics for continuous programs let $f_\omega : \mathbb{R}^n \to \mathbb{R}^n$ be the vector field defined by $f_\omega(\mathsf{a}) = \omega_{\mathsf{x}}^{\mathsf{a}}[\![\mathsf{v}]\!]\restriction_{\mathsf{x}}$ and define the state-flow map $\Psi_\omega(t) = \omega_{\mathsf{x}}^{\Phi_{f_\omega}(\omega\restriction_{\mathsf{x}}, t)}$. Now consider a state $\omega \in [\![\forall \mathsf{x}\, (\rho^\bullet \to |\mathsf{x}| \leq k)]\!]$ and let $K = \omega[\![k]\!]$. Clearly $\Psi_{f_\omega}(t)[\![k]\!] = K$ for all t in the domain.

For soundness of the $\to$ implication, assume $\omega \in [\![[\mathsf{x}' = \mathsf{v}\,\&\,\rho^\bullet \wedge |\mathsf{x}| \leq k]\varphi^\diamond]\!]$ and let

$$t = \inf\{s \leq K : (\omega\restriction_{\mathsf{x}}, s) \notin \Omega_{f_\omega} \text{ or } \Psi_\omega(s) \notin [\![\rho^\bullet \wedge |\mathsf{x}| \leq k]\!]\}.$$

There are two cases: either $t = \infty$ or $t \leq K$.

In case $t = \infty$, note that $(\omega, K) \in \Omega_{f_\omega}$ and $\Psi_\omega([0,K]) \in [\![\rho^\blacktriangle \wedge |\mathsf{x}| \leq k]\!]$. Thus, by assumption $\Psi_\omega([0,K]) \in [\![\varphi^\diamond]\!]$. Since the interval of solution existence is open, there is some $s > K$ such that $(\omega, s) \in \Omega_{f_\omega}$. Since the differential equation includes $\tau' = 1$ then also $\Psi_\omega([0,s]) \subseteq [\![\varphi^\diamond \vee |\mathsf{x}| > k]\!]$ and $\Psi_\omega(s) \in [\![|\mathsf{x}| > k]\!]$. Hence $\omega \in [\![\langle \mathsf{x}' = \mathsf{v} \,\&\, \varphi^\diamond \vee |\mathsf{x}| > k\rangle(\neg\rho^\blacktriangle \vee |\mathsf{x}| > k)]\!]$.

In case $t \leq K$ note that $\Psi_\omega([0,t)) \in [\![\rho^\blacktriangle \wedge |\mathsf{x}| \leq k]\!]$. (If instead t where not in the existence interval, the solution would have escaped $|\mathsf{x}| \leq k$ before then.) By continuity of Ψ_ω as $[\![\varphi^\blacktriangle \wedge |\mathsf{x}| \leq k]\!]$ is closed by Theorem 2 also $\Psi_\omega([0,t]) \in [\![\rho^\blacktriangle \wedge |\mathsf{x}| \leq k]\!]$. So by the assumption $\Psi_\omega([0,t]) \in [\![\varphi^\diamond]\!]$. By continuity of the flow and openness of $[\![\varphi^\diamond]\!]$ (Theorem 2), there is some $s > t$ such that $\Psi_\nu([0,s]) \in [\![\varphi^\diamond]\!]$. By definition of t without loss of generality $\Psi_\omega(s) \notin [\![\rho^\blacktriangle \wedge |\mathsf{x}| \leq k]\!]$. Thus $\omega \in [\![\langle \mathsf{x}' = \mathsf{v} \,\&\, \varphi^\diamond \vee |\mathsf{x}| > k\rangle(\neg\rho^\blacktriangle \vee |\mathsf{x}| > k)]\!]$.

For soundness of the $\leftarrow$ implication, assume

$$\omega \in [\![\langle \mathsf{x}' = \mathsf{v} \,\&\, \varphi^\diamond \vee |\mathsf{x}| > k\rangle(\neg\rho^\blacktriangle \vee |\mathsf{x}| > k)]\!]$$

Then there is s such that $\Psi_\omega([0, \min\{K, s\}]) \subseteq [\![\varphi^\diamond]\!]$ and either $\|\Psi_\omega(s)\| > T$ or $\Psi_\omega(s) \notin [\![\rho^\blacktriangle]\!]$. It follows immediately that $\omega \in [\![[\mathsf{x}' = \mathsf{v} \,\&\, \rho^\blacktriangle \wedge |\mathsf{x}| \leq k]\varphi^\diamond]\!]$. □

Proof of Theorem 4. By |&| it suffices to show equivalently

$$\begin{aligned} & \langle \mathsf{x}' = \mathsf{v} \,\&\, \rho^\diamond \wedge |\mathsf{x}| < k\rangle\varphi^\diamond \\ \leftrightarrow \quad & \exists m, \ell > 0 \, (\mathrm{bound}^\diamond \wedge \exists h > 0 \, \langle \varepsilon := 0; (?\rho^\diamond_{\mathsf{x}\pm\varepsilon}; \mathrm{step}^\diamond)^*\rangle(\varphi^\diamond_{\mathsf{x}\pm\varepsilon} \wedge \varepsilon < 1)) \end{aligned}$$

As in the definition of the semantics of continuous programs let $f_\omega : \mathbb{R}^n \to \mathbb{R}^n$ be $f_\omega(\mathsf{a}) = \omega^{\mathsf{a}}_{\mathsf{x}}[\![\mathsf{v}]\!]\restriction\mathsf{x}$ and let $\Psi_\omega(t) = \omega^{\Phi_{f_\omega}(\omega\restriction\mathsf{x},t)}_{\mathsf{x}}$ be the state-flow. Fix a state ω and note that $K := \omega[\![k]\!]$ is constant along the evolution.

For the forward implication, suppose there is t such that $\Psi_\omega(t) \in [\![\varphi^\diamond]\!]$, $\Psi_\omega([0,t]) \subseteq [\![\rho^\diamond]\!]$ and $\|\Psi_\omega(s)\|_\infty < K$ for all $s \in [0,t]$. Because $[\![\varphi^\diamond]\!]$ and $[\![\rho^\diamond]\!]$ are open by Theorem 2, without loss of generality, there is $\delta < 1$ such that

$$\begin{aligned} & \{\omega^{\mathsf{a}}_{\mathsf{x}} : \|\mathsf{a} - \Phi_{f_\omega}(\omega\restriction\mathsf{x}, t)\|_\infty < \delta\} \subseteq [\![\varphi^\diamond]\!] \\ & \{\omega^{\mathsf{a}}_{\mathsf{x}} : s \in [0,t] \,,\, \|\mathsf{a} - \Phi_{f_\omega}(\omega\restriction\mathsf{x}, s)\|_\infty < \delta\} \subseteq [\![\rho^\diamond]\!] \end{aligned}$$

Then let

$$M > \sup_{\|x\|_\infty \leq k+2} \|f_\omega(x)\|_\infty \quad \text{and} \quad L > \sup_{\|x\|_\infty \leq k+2} \|Df_\omega(x)\|_\infty \tag{1}$$

so that $\omega^{M,L}_{m,\ell} \in [\![\mathrm{bound}^\diamond]\!]$. By the convergence of Euler's method, there is a step size $H > 0$ such that the sum of the local approximation errors (accumulated in ε) in time t is bounded by δ. The Euler approximation with step size H then witnesses

$$\omega^{M,L,H}_{m,\ell,h} \in [\![\langle \varepsilon := 0; (?\rho^\diamond_{\mathsf{x}\pm\varepsilon}; \mathrm{step}^\diamond)^*\rangle(\varphi^\diamond_{\mathsf{x}\pm\varepsilon} \wedge \varepsilon < 1)]\!].$$

For soundness of the backward implication, suppose there are M and L satisfying (1). Moreover, assume that there is a step size $h > 0$ and a number of

iterations n, such that the n-step Euler iteration has an error bounded by $\delta < 1$, the δ-neighborhood of each Euler step remains in $[\![\rho^\diamond]\!]$ and the δ-neighborhood of the n-th Euler step is in $[\![\varphi^\diamond]\!]$. If the solution to the differential equation exists for time $t = nh$, then convergence of Euler's method ensures that $\Psi_\omega(t) \in [\![\varphi^\diamond]\!]$, $\Psi_\omega([0,t]) \subseteq [\![\rho^\diamond]\!]$ and $\|\Psi_\omega(s)\|_\infty \leq K$.

It remains to show that the solution exists for time $t = nh$. Assume for a contradiction that this is not the case. Then there must be $m < n$ such that the flow leaves the K-bounded region by time $s = mh < t$, i.e. $\|\Phi_{f_\omega}(\omega\restriction_{\mathsf{x}}, s)\|_\infty > K + 1$. Assume without loss of generality $\|\Phi_{f_\omega}(\omega\restriction_{\mathsf{x}}, s')\|_\infty < K + 2$ for all $s' \in [0, s]$. As the bounds of (1) apply in the $K + 2$ bounded region, the first m-steps of the Euler iteration are within distance δ of the actual solution. This is a contradiction, as $\delta < 1$ and the Euler iteration steps remain in the region bounded by K by the definition of $\text{step}^\diamond$. □