LIPIcs Leibniz International Proceedings in Informatics

Deductive Verification of Continuous Dynamical Systems

Ankur Taly[1], **Ashish Tiwari**[2]*

[1] Computer Science Dept., Stanford University
ataly@stanford.edu

[2] SRI International, Menlo Park, CA 94025
tiwari@csl.sri.com

ABSTRACT. We define the notion of inductive invariants for continuous dynamical systems and use it to present inference rules for safety verification of polynomial continuous dynamical systems. We present two different sound and complete inference rules, but neither of these rules can be effectively applied. We then present several simpler and practical inference rules that are sound and relatively complete for different classes of inductive invariants. The simpler inference rules can be effectively checked when all involved sets are semi-algebraic.

1 Introduction

The deductive rule for safety verification of sequential and concurrent programs was an important milestone in the field of formal program verification [6, 10, 12]. A program can be proved safe by constructing an inductive invariant that is strong enough to prove safety. Programs can be formally viewed as discrete state transition systems. If the predicate $t(\vec{x}, \vec{y})$ states that there is a discrete transition from the state $\vec{x}$ to the state $\vec{y}$ in the discrete state transition system $\mathtt{DTS}$, and if $\mathtt{Init}$ and $\mathtt{Safe}$ are, respectively, the initial states of $\mathtt{DTS}$ and the hypothesized safe set, then the classical inference rule for safety verification is given as follows:

$$\begin{array}{llrcl}
(1) & \forall \vec{x} & \vec{x} \in \mathtt{Init} & \Rightarrow & \vec{x} \in \mathtt{Inv} \\
(2) & \forall \vec{x}, \vec{y} & \vec{x} \in \mathtt{Inv} \wedge t(\vec{x}, \vec{y}) & \Rightarrow & \vec{y} \in \mathtt{Inv} \\
(3) & \forall \vec{x} & \vec{x} \in \mathtt{Inv} & \Rightarrow & \vec{x} \in \mathtt{Safe} \\
\hline
 & & \mathtt{Reach}(\mathtt{DTS}) & \subseteq & \mathtt{Safe}
\end{array}$$

This rule essentially says that we can prove that all reachable states of $\mathtt{DTS}$ lie inside the safe set $\mathtt{Safe}$ by finding a suitable "inductive invariant" $\mathtt{Inv}$.

A valuable property of the deductive verification rule is that it is both sound and complete. Soundness here means that if a program is proved correct using the rule, then that program indeed satisfies the safety property. Completeness means that if the given program is actually safe, then there is an inductive invariant $\mathtt{Inv}$ that satisfies the Conditions (1), (2) and (3) of the deductive verification rule. The above rule, however, applies only to discrete state transition systems where the "next" states can be effectively specified.

While *discrete state transition systems* is a powerful modeling formalism, it is inadequate for modeling systems that involve physical components. Physical systems are typically

*Research supported in part by NSF grants CNS-0720721 and CSR-0917398 and NASA grant NNX08AB95A.

Foundations of Software Technology and Theoretical Computer Science (Kanpur) 2009.
Editors: Ravi Kannan and K. Narayan Kumar; pp 383–394
Leibniz International Proceedings in Informatics (LIPIcs), Schloss Dagstuhl - Leibniz-Zentrum für Informatik, Germany.
Digital Object Identifier: 10.4230/LIPIcs.FSTTCS.2009.2334

modeled using differential equations as *continuous dynamical systems*. The formalisms of continuous dynamical systems and discrete transition systems can be combined to give *hybrid dynamical systems*. Hybrid systems are immensely useful in describing systems that have physical and computational components, such as embedded and control systems, as well as, systems that operate at multiple different time scales, such as biological systems.

This paper presents a deductive verification rule for continuous dynamical systems. When combined with the above rule for discrete state transition systems, we get a deductive verification rule for hybrid systems. The challenge in coming up with a deductive verification rule for continuous dynamical systems is that there is no useful notion of a "next" state. In this paper, we use "continuity" to formulate the deductive verification rule. One of the technical difficulties here is to obtain a rule that is simultaneously (1) sound, (2) complete, and (3) effectively checkable. It is easy to propose rules that compromise one or more of these three requirements. The main results of this paper are (a) two distinct sound and complete rules, but these are not directly checkable, and (b) three simpler sound and effectively checkable rules, that are relatively complete for large and useful classes of systems and invariants.

Motivation and Related Work. From a purely theoretical perspective, it is appealing to have an effective, sound, and relatively complete inference rule for safety verification of continuous systems. Recently, however, promising practical techniques have been proposed for safety verification that are directly based on using such inference rules. One such technique – that is especially effective for safety verification of continuous and hybrid systems – is *bounded verification*. Bounded verification is the dual of bounded model checking. Whereas bounded model checking searches for a bounded counter example for safety, bounded verification searches for a bounded proof for safety. The essential idea in bounded verification is to search for an inductive invariant of a given form. Note that the inference rule for safety verification requires proving the formula

$$\exists \mathtt{Inv} : \forall \vec{x}, \vec{y} : \phi(\mathtt{Inv}, \vec{x}, \vec{y}), \tag{1}$$

where ϕ is simply a conjunction of Formulas (1), (2) and (3) from the rule above. This formula involves a second-order quantification. We can eliminate this second-order quantification by restricting the form of the inductive invariant `Inv`. For example, assuming `Inv` can be written as $\psi(\vec{u}, \vec{x})$, over some unknown parameters $\vec{u}$, Formula 1 changes to

$$\exists \vec{u} : \forall \vec{x}, \vec{y} : \phi(\psi(\vec{u}, \vec{x}), \vec{x}, \vec{y}). \tag{2}$$

Formula 2 is now a first-order $\exists\forall$ formula. If this formula is valid, then we know there is an inductive invariant that proves safety. Further details on bounded verification, can be found in the work of Gulwani et al. [7] and Gulwani and Tiwari [8].

The formula $\psi(\vec{u}, \vec{x})$ can be seen as a template for the invariant. The idea of using templates is not new. In fact, it is the classical approach used to prove stability in control theory. Recently, it has also been used for safety verification for discrete programs [2, 7, 11, 18] and continuous and hybrid systems [17, 15, 1, 20, 8]. These papers use templates for performing bounded verification, but differ in the details about their use of the inference rule to construct ϕ' in Formula 2 and their use of the constraint solving technique to solve the

$\exists\forall$ constraint. Since template-based verification is not the main topic of this paper, but just used as a motivation, we do not discuss the related literature here. However, the inference rules used in the papers on verification of continuous and hybrid systems are relevant to the work in this paper and we discuss them briefly here and in the rest of the article.

If the hypothesized invariant $\texttt{Inv}$ is a polynomial equation, $p = 0$, then there is a simple way to check invariance: whenever $p = 0$, the time derivative of p, $\frac{dp}{dt}$, should also be 0. This verification rule for equational invariants was used by Sankaranarayanan et al. [17]. If the invariant is an inequality, such as $p \geq 0$, then there are several sufficient checks, such as, $\frac{dp}{dt} \geq 0$ whenever $p \geq 0$. This test is very strong: it requires that p is increasing everywhere inside the invariant set. This sound, but incomplete, test has been used by Platzer et al. [14, 13]. We can weaken the test, and check $\frac{dp}{dt} \geq 0$ only on points where $p = 0$ [15, 8], but this variant is not sound in general. This is discussed in detail later.

Outline of the Paper. We formally define continuous dynamical systems in Section 2 and present two distinct sound and complete deductive verification rules for continuous dynamical systems in Section 3. In Section 4, we first present inference rules that are interesting from a practical point of view and compromise either soundness or completeness. We then present three sound and relatively complete inference rules.

2 Continuous Dynamical System

DEFINITION 1.*[Continuous Dynamical System] A* continuous dynamical system $\texttt{CDS}$ *is a tuple* $(X, \texttt{Init}, f)$ *where* X *is a finite set of variables interpreted over the reals* $\mathbb{R}$, $\mathbf{X} = \mathbb{R}^X$ *is the set of all valuations of the variables* X, $\texttt{Init} \subseteq \mathbf{X}$ *is the set of initial states, and* $f : \mathbf{X} \mapsto \mathbf{X}$ *is a vector field that specifies the continuous dynamics.*

Note that $\mathbb{R}^{\mathrm{X}}$ is isomorphic to the n-dimensional real space $\mathbb{R}^n$ where $n = |\mathrm{X}|$ is the number of variables in X. Note also that the continuous dynamical systems we consider here are autonomous, that is, they have no inputs. We assume that f is Lipshitz, which guarantees that the ordinary differential equations $\frac{d\mathrm{X}}{dt} = f(\mathrm{X})$ have a unique solution. In fact, the following property [4] of Lipschitz vector fields will be used in the proofs.

PROPOSITION 2.*[Theorem 2.3.1, p80 [4]] Consider a Lipschitz vector field* f *and the initial value problem* $\frac{dX(t)}{dt} = f(X(t))$, $X(0) = \vec{x}_0$. *The solution of this problem, denoted by* $F(\vec{x}_0, t)$, *always exists and is unique. Moreover,* $F(\vec{x}_0, t)$ *depends continuously on the initial state* $\vec{x}_0$.

The meaning of a continuous dynamical system is simply the collection of all possible trajectories starting from an initial state. Formally, if $F(\vec{x}_0, t)$ is the solution of $\frac{d\mathrm{X}(t)}{dt} = f(\mathrm{X}(t))$, $\mathrm{X}(0) = \vec{x}_0$, then the semantics, $[[\texttt{CDS}]]$, of a continuous dynamical system $\texttt{CDS} = (\mathrm{X}, \texttt{Init}, f)$ is given as

$$[[\texttt{CDS}]] \;:=\; \{F_1 : [0,\infty) \mapsto \mathbf{X} \;\mid\; F_1(t) = F(\vec{x}_0, t), \quad \vec{x}_0 \in \texttt{Init} \;\}$$

The above semantics using flow functions is broadly referred to as the *flow semantics* [21]. One can also give a *transition semantics* using discrete state transition systems [9], but the distinction [5] is not relevant for this paper.

The set of reachable states for a continuous dynamical system CDS, Reach(CDS), is given by $\{\vec{x} \in \mathbf{X} \mid \exists F \in [[\texttt{CDS}]], \exists t \geq 0 : \vec{x} = F(t)\}$. A (safety) property, Safe, is simply a subset of the state space $\mathbf{X}$. A property Safe is an *invariant* (for the system CDS) if Reach(CDS) $\subseteq$ Safe. We are interested in solving the following problem in this paper:

DEFINITION 3. *[Safety Verification Problem] Given a continuous dynamical system* CDS *and a safety property* Safe*, determine if* Safe *is an invariant for* CDS*.*

One of the classical methods to solve the safety verification problem is based on finding stronger invariants that are also *inductive*. By introducing the extra requirement of inductiveness, the "global" test for invariance, viz. *all* reachable states are contained in Safe, reduces to a simpler "local" test, viz. every *single* transition out of Safe state goes into only a Safe state.

3 Sound and Complete Rules

In this section we present two verification rules for solving the problem described in Definition 3. Each rule replaces the global test for invariance by a local test for inductiveness.

We fix our notation and denote the given continuous dynamical system by CDS $=$ (X, Init, f) and the given safety property by Safe. The challenge in defining a local inductiveness test is that, for continuous dynamical systems, there is no clear notion of a "next" state in the flow semantics. Even if we use the transition semantics, the set of all the uncountably many next states is equal to the Reach set and hence the distinction between inductive invariants and general invariants is lost. However, using continuity, instead of using arbitrary future states, we can look at only states reachable in an ϵ-future and require that they remain inside Inv. This is formalized below in $(\mathcal{A}2)$.

DEFINITION 4. *[Inductive Invariant] A set* $Inv \subset \mathbb{R}^X$ *is an* inductive invariant *for a given continuous dynamical system* $CDS := (X, Init, f)$ *if the following conditions hold:*

$$(\mathcal{A}1) \quad Init \subseteq Inv$$
$$(\mathcal{A}2) \quad \forall \vec{x} \in Inv : \exists t_0 > 0 : \forall 0 \leq t < t_0 : F(\vec{x}, t) \in Inv$$

where F *is the solution of the initial value problem* $\frac{dX(t)}{dt} = f(X(t)), X(0) = \vec{x}$.

A closed set that is an inductive invariant in the above sense contains all the reachable states and hence it is indeed an invariant.

PROPOSITION 5. *Let* Inv *be a closed inductive invariant for the continuous dynamical system* $CDS := (X, Init, f)$*. Then,* $Reach(CDS) \subseteq Inv$*.*

However, Definition 4 is not directly useful for checking inductiveness because (a) it uses quantifier alternation ($\forall\exists\forall$) and (b) it uses the solution F of the differential equations. For most interesting applications, it may be difficult, if not impossible, to compute F analytically. Fortunately, there are two different ways in which we can check for inductiveness without using F. Before describing them, we first concretize the specification language for CDS and Safe.

$$(S1)\quad \texttt{Init}(\vec{x}) \Rightarrow p(\vec{x}) \geq 0 \qquad (T1)\quad \texttt{Init}(\vec{x}) \Rightarrow p(\vec{x}) \geq 0$$

$$(S2)\quad p(\vec{x}) = 0 \Rightarrow f(\vec{x}) \in T(p \geq 0)(\vec{x}) \qquad (T2)\quad p = 0 \Rightarrow (\bigwedge_{i=1}^{k-1} L_f^{(i)}(p) = 0 \Rightarrow L_f^{(k)}(p) \geq 0) \text{ for } k = 1, 2, \ldots$$

$$(S3)\quad \frac{p(\vec{x}) \geq 0 \Rightarrow \texttt{Safe}(\vec{x})}{\texttt{Reach}(\texttt{CDS}) \subseteq \texttt{Safe}} \qquad (T3)\quad \frac{p(\vec{x}) \geq 0 \Rightarrow \texttt{Safe}(\vec{x})}{\texttt{Reach}(\texttt{CDS}) \subseteq \texttt{Safe}}$$

Figure 1: Inference rules for safety verification of continuous system $\texttt{CDS} := (\texttt{X}, \texttt{Init}, f)$ and safety property $\texttt{Safe} \subseteq \mathbf{X}$.

Since we are interested in computability, henceforth, we assume that the continuous dynamical system $\texttt{CDS} := (\texttt{X}, \texttt{Init}, f)$ and the safe set $\texttt{Safe}$ are specified using polynomials. Let $L := \{\mathbb{Q}, +, -, *, \geq, >, =\}$ be a language containing all rational constants $\mathbb{Q}$, function symbols $+, -, *$ and predicates $\geq, >, =$. These symbols are interpreted over the reals in the usual way. We fix $\texttt{X}$ to be the set of variables. A term over variables $\texttt{X}$ will just be a polynomial in the ring $\mathbb{Q}[\texttt{X}]$. Atomic formulas consist of polynomial equalities and inequalities. A set $\texttt{S} \subseteq \mathbb{R}^n$ is *semi-algebraic* if it represents the solutions of a (quantifier-free) formula. A $\texttt{CDS} := (\texttt{X}, \texttt{Init}, f)$ is a *polynomial CDS* if $\texttt{Init}$ is semi-algebraic and the vector field is specified using only polynomials from $\mathbb{Q}[\texttt{X}]$.

For simplicity of presentation, we will initially restrict the set $\texttt{Inv}$ to be of the form $p \geq 0$ for some polynomial p. We will later extend the results to boolean combinations. Since we are restricting $\texttt{Inv}$ to be in a certain class, we will lose completeness. However, we are interested in "relative completeness"; that is, if there is an inductive invariant in the restricted class, then the deductive verification rule should be applicable.

We are now ready to present the two different ways for checking inductiveness without using F. First, we use a result in Control Theory, called Nagumo's theorem, that says that a set $\texttt{Inv}$ is an invariant only if, at every point $\vec{x}$ on the boundary of $\texttt{Inv}$, the vector field $f(\vec{x})$ at that point points "inwards". Formally, the set of vectors that point "inwards" at point $\vec{x}$ define the tangent cone at $\vec{x}$.

DEFINITION 6. *[Tangent Cone, Definition 3.1 in [3]] Let $S \subset \mathbb{R}^n$ be a closed set. Let $\vec{x} \in \mathbb{R}^n$. The tangent cone to S at $\vec{x}$ is the set*

$$T(S)(x) \quad := \quad \{\vec{z} \in \mathbb{R}^n \mid \liminf_{\alpha \to 0} \frac{d(\vec{x} + \alpha\vec{z}, S)}{\alpha} = 0\} \tag{3}$$

where $d(\vec{x}, S) := \inf_{\vec{y} \in S} ||\vec{x} - \vec{y}||$ is the distance of $\vec{x}$ from S and $||\cdot||$ is any norm in $\mathbb{R}^n$.

Figure 1 (Left) presents an inference rule for safety verification of continuous systems. Note that Condition (S2) says that for every point on the boundary of $\texttt{Inv}$, the vector field f is in the tangent cone at that point. Nagumo's theorem states that for closed sets $\texttt{Inv}$, Condition (S2) from Figure 1 is equivalent to Condition $(\mathcal{A}2)$ from Definition 4. We refer the reader to the review article by Blanchini for details [3].

The key idea behind the second approach for automating the test of Condition $(\mathcal{A}2)$ is the use of *Lie* derivatives. Intuitively, we can check that trajectories do not leave $p \geq 0$ by

checking that $\frac{dp}{dt}$ is greater-than zero whenever $p = 0$. Technically, the derivative of p with respect to time, $\frac{dp}{dt}$, is called the *Lie derivative*, $L_f(p)$, of p with respect to the vector field f. It can be computed using the chain rule, as shown below. Let us define the notation $L_f^{(n)}(p)$ to denote the n-th derivative of p with respect to time. Formally,

$$L_f^{(n)}(p) \;:=\; \begin{cases} \sum_{x \in \mathrm{X}} \frac{\partial p}{\partial x}\frac{dx}{dt} \;:=\; \vec{\nabla} p \cdot f := (\frac{\partial p}{\partial x_1}, \frac{\partial p}{\partial x_2}, \ldots) \cdot (\frac{dx_1}{dt}, \frac{dx_2}{dt}, \ldots) & \text{if } n = 1 \\ \frac{dL_f^{(n-1)}(p)}{dt} & \text{otherwise} \end{cases} \tag{4}$$

where the time-derivative, $\frac{d}{dt}$, is always computed using the chain rule as $\frac{dg}{dt} = \vec{\nabla} g \cdot f$. If f is specified using polynomials (i.e., $\frac{dx}{dt}$ is a polynomial for every variable x) and if p is a polynomial in $\mathbb{Q}[\mathrm{X}]$, then Equation 4 shows that $L_f^{(n)}(p)$ is a polynomial in $\mathbb{Q}[\mathrm{X}]$ and it can be symbolically computed. The second inference rule for checking inductiveness is shown in Figure 1(Right). Note that Condition (T2) requires that, *for all* k, the k-th derivative be non-negative whenever the first $k-1$ derivatives are zero.

The two deductive verification rules given in Figure 1 are both sound and (relatively) complete. For lack of space, proofs are not provided.

THEOREM 7. *[Soundness] Let* $\mathtt{CDS} := (\mathtt{X}, \mathtt{Init}, f)$ *be a continuous dynamical system and* $\mathtt{Safe} \subseteq \mathbf{X}$ *be a safety property. If there is a set* $\mathtt{Inv}$ *that satisfies Conditions (S1), (S2) and (S3) from Figure 1(Left), or alternatively, it satisfies Conditions (T1), (T2) and (T3) from Figure 1(Right), then* $\mathtt{Reach}(\mathtt{CDS}) \subseteq \mathtt{Safe}$.

THEOREM 8. *[Relative Completeness] Let* $\mathtt{CDS} := (\mathtt{X}, \mathtt{Init}, f)$ *be a CDS and* $\mathtt{Safe}$ *be a closed set such that* $\mathtt{Reach}(\mathtt{CDS}) \subseteq \mathtt{Safe}$. *If there is an inductive invariant* $p \geq 0$ *such that* $p \geq 0 \Rightarrow \mathtt{Safe}$, *then* $p \geq 0$ *also satisfies Conditions (S1), (S2) and (S3) from Figure 1(Left), as well as, Conditions (T1), (T2) and (T3) from Figure 1(Right).*

The inference rules in Figure 1 can be generalized to also handle Boolean combinations of predicates of the form $p \geq 0$ (see Discussion in Section 4) and these generalized rules will be complete for all semi-algebraic invariants.

Comparing the two inference rules. Since the two sets of conditions in Figure 1 are both sound and relatively complete for showing inductive invariance, it is tempting to assume that they are "essentially the same". These two tests are indeed "globally equivalent": if every point on the boundary satisfies Condition (S2), then every point on the boundary also satisfies Condition (T2), and vice-versa. However, the two tests are distinct tests and they are *not* "locally equivalent"; that is, they may disagree on individual points.

Example 1 *Consider the constant vector field* $f((x,y)) = (1,0)$ *and consider the candidate invariant region,* $-x^2 - y^2 + 2y \geq 0$. *The candidate invariant set is a circle of radius* 1 *centered at* $(0,1)$ *and hence clearly the vector field is tangential to the invariant set at the origin; that is,* $(1,0) \in T(-x^2 - y^2 + 2y \geq 0)((0,0))$. *Hence Condition (S2) evaluates to true for point* $(0,0)$. *However, the derivative test fails at* $(0,0)$*: though* $\frac{dp}{dt}$ *at* $(0,0)$ *is* 0, *the second derivative is negative (everywhere):* $\frac{dp}{dt} = -2x\frac{dx}{dt} - (2y-2)\frac{dy}{dt} = -2x$, $\frac{d^2p}{dt^2} = -2\frac{dx}{dt} = -2$. *This shows that Condition (T2) fails at* $(0,0)$. *Thus, Condition (S2) and Condition (T2) give different answers at the point* $(0,0)$. *However, they both agree globally that the candidate invariant set here is not an invariant.*

$$\begin{array}{ll} (A1) & \texttt{Init}(\vec{x}) \Rightarrow p(\vec{x}) \geq 0 \\ (A2) & p(\vec{x}) = 0 \Rightarrow L_f(p)(\vec{x}) \geq 0 \\ (A3) & \underline{p(\vec{x}) \geq 0 \Rightarrow \texttt{Safe}(\vec{x})} \\ & \texttt{Reach(CDS)} \subseteq \texttt{Safe} \end{array} \qquad \begin{array}{ll} (B1) & \texttt{Init}(\vec{x}) \Rightarrow p(\vec{x}) \geq 0 \\ (B2) & p(\vec{x}) = 0 \Rightarrow L_f(p)(\vec{x}) > 0 \\ (B3) & \underline{p(\vec{x}) \geq 0 \Rightarrow \texttt{Safe}(\vec{x})} \\ & \texttt{Reach(CDS)} \subseteq \texttt{Safe} \end{array}$$

Figure 2: An unsound, but relatively complete, rule (left) and a sound, but incomplete, rule (right) for safety verification of polynomial CDS $\texttt{CDS} := (\texttt{X}, \texttt{Init}, f)$ and safety property $\texttt{Safe} \subseteq \mathbf{X}$.

Although the verification rules in Figure 1 are both sound and relatively complete, they are not computationally feasible as there is no easy way to verify Condition (S2) and Condition (T2): the former involves reasoning about the tangent cone, whereas the latter is an infinite set of conditions. We will next present computable conditions and prove their soundness or completeness by comparing them to Condition (S2) or Condition (T2).

4 Practical Rules for Safety Verification of Polynomial CDS

In this section, we present inference rules that can be applied in practice for performing safety verification of continuous systems. We shall also point to the literature where these rules have been used. The rules will compromise either soundness or completeness.

Figure 2 presents two approximations of the inference rule in Figure 1(Right). First, instead of performing the infinitely many checks in Condition (T2)– one for each k – we can just perform the check for $k = 1$ and ignore the other checks. This gives an unsound, but relatively complete, inference rule, shown in Figure 2(Left). The following example shows the unsoundness and was mentioned to us by Andre Platzer.

Example 2 *Consider the system* $\texttt{CDS} := (\{x\}, \{x = 0\}, f)$ *where* $f(x) = 1$ *and the safety property* $-x^2 \geq 0$. *Since initially* $x = 0$ *and since* $\frac{dx}{dt} = f(x) = 1$, x *takes positive values and hence the safety property is violated. However, the rule in Figure 2(Left) can be applied successfully using* $-x^2$ *as* p. *Condition (A2) is verified because the following is a theorem in the theory of reals:* $-x^2 = 0 \Rightarrow -2x * 1 \geq 0$. *This example shows that the rule in Figure 2(Left) is unsound.*

Example 2 suggests that we can regain soundness by replacing the check $L_f(p) \geq 0$ by the stronger test $L_f(p) > 0$. This gives us the inference rule in Figure 2(Right). However, we lose completeness.

Example 3 (Incompleteness) *Consider the system* $\texttt{CDS} := (\{x\}, \{x = 0\}, f)$ *where* $f(x) = 0$ *and the safety property* $x \geq 0$. *Since initially* $x = 0$ *and since* $\frac{dx}{dt} = f(x) = 0$, *clearly* CDS *is safe with respect to the given safety property. In fact, there is an inductive invariant* $x \geq 0$ *(of the form* $p \geq 0$*) that can prove this safety property. However, the rule in Figure 2 fails: for any* $p \in \mathbb{Q}[x]$, $L_f(p)$ *is always* 0, *and it is never strictly positive (as required by Condition (B2)).*

The rules in Figure 2 are commonly used. Despite the unsoundness, the inference rule in Figure 2(Left) has been used in the work by Gulwani and Tiwari [8] and Prajna and Jadbabaie [15]. The sound, but incomplete, variant in Figure 2(Right) has been used by Prajna, Jadbabaie and Pappas [16].

$$\frac{(C1)\ \ \texttt{Init} \Rightarrow p \geq 0 \quad (C2)\ \ p = 0 \Rightarrow L_f(p) \geq 0 \quad (C2')\ \ p = 0 \Rightarrow \vec{\nabla} p \neq 0 \quad (C3)\ \ p \geq 0 \Rightarrow \texttt{Safe}}{\texttt{Reach}(\texttt{CDS}) \subseteq \texttt{Safe}}$$

$$\frac{(D1)\ \ \texttt{Init} \Rightarrow p \geq 0 \quad (D2)\ \ p = 0 \Rightarrow L_f(p) \geq 0 \quad (D2')\ \ p = 0 \wedge \vec{\nabla} p = 0 \Rightarrow \neg \texttt{neg}(p, \vec{\mathrm{x}}, f(\vec{\mathrm{x}})) \quad (D3)\ \ p \geq 0 \Rightarrow \texttt{Safe}}{\texttt{Reach}(\texttt{CDS}) \subseteq \texttt{Safe}}$$

Figure 3: Sound inference rules for safety verification of polynomial CDS $\texttt{CDS} := (\mathrm{X}, \texttt{Init}, f)$ and safety property $\texttt{Safe} \subseteq \mathbf{X}$ that are also complete for a certain class of invariants.

Inference Rule Complete for Smooth Invariants

The case that leads to unsoundness or incompleteness is when $p(\vec{\mathrm{x}}) = 0$ and $L_f(p)(\vec{\mathrm{x}}) = 0$. Intuitively, one expects that the condition $L_f(p)(\vec{\mathrm{x}}) = 0$ should hold only when the vector field is "tangential" to the invariant set $p \geq 0$. Unfortunately, it also holds in some degenerate cases. One such degenerate case is when $\vec{\nabla} p = 0$. The inference rule in Figure 3(Left) explicitly rules out such cases. Let us say that the boundary of a set $p \geq 0$ is *smooth* if, $\vec{\nabla} p(\vec{\mathrm{u}}) \neq 0$ for all points $\vec{\mathrm{u}}$ s.t. $p(\vec{\mathrm{u}}) = 0$. Condition (C2′) in Figure 3(Left) explicitly checks that the boundary of the invariant set is smooth. With this additional check, the inference rule in Figure 3(Left) can be shown to be sound.

Example 4 *Consider the dynamical system from Example 2. We cannot use the rule in Figure 3 on it. If we use $-x^2$ as p, Condition (C2′) becomes $-x^2 = 0 \Rightarrow -2x \neq 0$ which is false over the reals.*

The inference rule in Figure 3(Left) is not only sound, but also complete for invariants $p \geq 0$ whose boundary is smooth. This is the case, for example, when p is linear, which is a particularly useful class [8]. Figure 3(Left) fails on invariants with non-smooth boundaries.

Example 5 *Consider the system $\texttt{CDS} := (\{x, y, z\}, \texttt{Init}, f)$, where $\texttt{Init}$ is the set $x^2 + y^2 \leq z^2$ and the vector field f is given by $f((x, y, z)) := (-x, -y, -z)$. Thus, at every point, the vector field points to the origin and the initial set is a cone. We wish to prove that the set $\texttt{Init}$ is safe; i.e., $\texttt{Safe} = \texttt{Init}$. The inductive invariant $z^2 - x^2 - y^2 \geq 0$ can prove safety. However, there is no polynomial p such that $p \geq 0$ satisfies Conditions (C1), (C2), (C2′) and (C3). Suppose p is such a polynomial. Then, since the set $\texttt{Init}$ is equal to the set $\texttt{Safe}$, the set $\texttt{Inv} := \{\vec{\mathrm{u}} \mid p(\vec{\mathrm{u}}) \geq 0\}$ has to be necessarily equal to these two sets (by Condition (C1) and (C3)). But then, Condition (C2′) will fail because at the boundary point $(0, 0, 0)$ the gradient of p cannot be nonzero.*

Inference Rule Complete for Quadratic Invariants

We can generalize Condition (C2′) to require that, at all points where $p = 0$ and $\vec{\nabla} p = 0$, the vector field f is "pointing inside" (Figure 3(Right)). Before we outline the test for "pointing inside", we need the following definition.

DEFINITION 9. *[Homogeneous decomposition, $\texttt{zero}, \texttt{pos}, \texttt{neg}$] A polynomial $p \in \mathbb{Q}[X]$ is a* homogeneous polynomial of degree *k if the total degree of each monomial in p is k. A* homogeneous decomposition *of p is obtained by writing p as $\sum_{i=1}^{n} p_i$, where p_i is homogeneous with degree k_i and $k_i < k_j$ for $i < j$. Let $p(\vec{x} + \vec{y})_i$ denote the i-th homogeneous*

$$(F1) \qquad \texttt{Init} \Rightarrow p \geq 0$$

$$(F2) \quad p = 0 \Rightarrow \neg \texttt{neg}(p, \vec{x}, f) \vee \bigvee_{k=2}^{n} (\texttt{kneg}(p, \vec{x}, f, k) \wedge \bigvee_{l<k} (\exists g : \texttt{pos}(p_l, f, g) \wedge \bigwedge_{j<l} \texttt{zero}(p_j, f, g)))$$

$$(F3) \qquad p \geq 0 \Rightarrow \texttt{Safe}$$

$$\texttt{Reach}(\texttt{CDS}) \subseteq \texttt{Safe}$$

Figure 4: Sound, and relatively complete, deductive rule for solving the safety verification problem for polynomial CDS $\texttt{CDS} := (\texttt{X}, \texttt{Init}, f)$ and safety property $\texttt{Safe} \subseteq \mathbf{X}$.

component of $p(\vec{x}+\vec{y})$ when viewed as a polynomial in $\vec{y}$ (with coefficients in $\mathbb{Q}[\vec{x}]$). The predicates $\texttt{zero}, \texttt{pos}, \texttt{neg}$ and $\texttt{kneg}$ are defined as follows:

$$\begin{aligned} \texttt{pos}(p, \vec{x}, \vec{u}) &:= \bigvee_{k=1}^{n} (p(\vec{x}+\vec{y})_k(\vec{u}) > 0 \wedge \bigwedge_{i=1}^{k-1} p(\vec{x}+\vec{y})_i(\vec{u}) = 0) \\ \texttt{kneg}(p, \vec{x}, \vec{u}, k) &:= (p(\vec{x}+\vec{y})_k(\vec{u}) < 0 \wedge \bigwedge_{i=1}^{k-1} p(\vec{x}+\vec{y})_i(\vec{u}) = 0) \end{aligned}$$

$$\texttt{zero}(p, \vec{x}, \vec{u}) := \bigwedge_{i=1}^{n} p(\vec{x}+\vec{y})_i(\vec{u}) = 0 \qquad \texttt{neg}(p, \vec{x}, \vec{u}) := \bigvee_{i=1}^{n} \texttt{kneg}(p, \vec{x}, \vec{u}, i)$$

If p is a polynomial and $\vec{x}, \vec{u}$ are two points such that $p(\vec{x}) = 0$, then
(a) $\texttt{pos}(p, \vec{x}, \vec{u})$ is equivalent to $\exists \alpha_0 > 0 : \forall (0 < \alpha \leq \alpha_0) : p(\vec{x} + \alpha\vec{u}) > 0$.
(b) $\texttt{zero}(p, \vec{x}, \vec{u})$ is equivalent to the fact that $p(\vec{x} + \alpha\vec{u}) = 0$ for all α.
(c) $\texttt{neg}(p, \vec{x}, \vec{u})$ is equivalent to $\exists \alpha_0 > 0 : \forall (0 < \alpha \leq \alpha_0) : p(\vec{x} + \alpha\vec{u}) < 0$.

Using the predicate $\texttt{neg}$, the inference rule in Figure 3(Right) checks that, for every point $\vec{x}$ such that $p(\vec{x}) = 0$ and $\vec{\nabla} p(\vec{x}) = 0$, it is the case that moving along the direction of the vector field $f(\vec{x})$ at the point $\vec{x}$, we move inside the invariant set $p \geq 0$. Figure 3(Right) generalizes the rule in Figure 3(Left). We will later see that it is complete for quadratic p.

Inference Rule Complete for Convex Invariants

Figure 4 presents an inference rule that generalizes the above two rules and can be shown to be complete for a larger class of invariants that includes linear, smooth and quadratic invariants. The rule in Figure 4 checks that for each point $\vec{x}$ on the boundary ($p(\vec{x}) = 0$), either we move inside the set $p \geq 0$ as we move from $\vec{x}$ along the vector field direction $f(\vec{x})$, or we move outside but there is a direction g such that if we go along g, we can make $p = 0$ "sufficiently quickly"; see illustration in Figure 5. The following example illustrates the notation from Definition 9 and the inference rule in Figure 4.

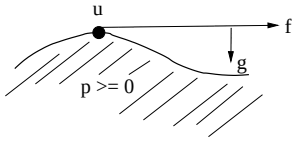

Figure 5: Illustration

Example 6 *Consider $\texttt{CDS} := (\{x_1, x_2\}, \texttt{Init}, f)$, where $\texttt{Init}$ is given by $x_1 = 2, x_2 = 0$ and $f(x_1) = x_2, f(x_2) = -x_1$. Let p be $-x_1^2 - x_2^2 + 4$. The set $p \geq 0$ is an inductive invariant of*

this `CDS`. *Let* $\vec{u}$ *be a point on the boundary; i.e.,* $p(\vec{u}) = 0$. *Moving the origin to* $\vec{u}$, *we get the new polynomial* $p(\vec{u}+\vec{x}) = -(u_1+x_1)^2 - (u_2+x_2)^2 + 4$ *which is equal to* $(-u_1^2 - u_2^2 + 4) - 2u_1x_1 - 2u_2x_2 - x_1^2 - x_2^2$. *Since* $p(\vec{u}) = 0$, *the new polynomial simplifies to* $-2u_1x_1 - 2u_2x_2 - x_1^2 - x_2^2$. *This has two homogeneous components:*

$$\begin{array}{lll} p_1 := p(\vec{u}+\vec{x})_1 := -2u_1x_1 - 2u_2x_2 & & \text{homogeneous with degree } k_1 = 1 \\ p_2 := p(\vec{u}+\vec{x})_2 := -x_1^2 - x_2^2 & & \text{homogeneous with degree } k_2 = 2 \end{array}$$

We now verify that $-x_1^2 - x_2^2 + 4 \geq 0$ *satisfies Condition (F2):*

$$p_1(f(\vec{u})) := -2u_1u_2 + 2u_2u_1 = 0 \qquad p_2(f(\vec{u})) := -u_1^2 - u_2^2$$

Since $p(\vec{u}) = 0$, *which is* $-u_1^2 - u_2^2 + 4 = 0$, *implies* $p_1(f(\vec{u})) = 0$ *and* $p_2(f(\vec{u})) < 0$, *we have* $\texttt{kneg}(p, \vec{u}, f(\vec{u}), 2)$ *holds. Clearly,* $\texttt{zero}(p, \vec{u}, f(\vec{u}))$ *and* $\texttt{pos}(p, \vec{u}, f(\vec{u}))$ *do not hold. Thus, we see that the direction* $f(\vec{u})$ *takes the point outside of the invariant set (as in Figure 5). However, Condition (F2) is true since for direction* $g := (-u_1, -u_2)$, $\texttt{pos}(p_1, f(\vec{u}), g)$ *holds:*

$$p_1(f(\vec{u})+\vec{x})_1 := -2u_1x_1 - 2u_2x_2, \qquad p_1(f(\vec{u})+\vec{x})_1(g) := 2u_1^2 + 2u_2^2 = 2*4 = 8 > 0$$

The rule in Figure 4 is complete for the class of invariants `Inv` that are `convex`.

DEFINITION 10. *The predicate* $\texttt{convex}(p \geq 0)$ *holds for the set* $p \geq 0$ *if, for any points* $\vec{u}$ *and* $\vec{v}$, *if* $p(\vec{u}) \geq 0$ *and* $p(\vec{v}) \geq 0$, *then* $p(\vec{u} + \alpha\vec{v}) \geq 0$ *for all* $0 \leq \alpha \leq 1$.

For example, the set $-x^2 - y^2 + 1 \geq 0$ is `convex`, but the set $-x^2 - y^2 + 1 = 0$, which can be encoded as $-(-x^2 - y^2 + 1)^2 \geq 0$, is not `convex`.

Soundness and Relative Completeness

THEOREM 11. *[Soundness] Let* $\texttt{CDS} := (X, \texttt{Init}, f)$ *be a CDS and* `Safe` *be a safety property. If* $p \in \mathbb{Q}[X]$ *is a polynomial that satisfies Conditions (C1), (C2), (C2') and (C3) of Figure 3(Left), or alternatively Conditions (D1), (D2), (D2') and (D3) of Figure 3(Right), or alternatively, Conditions (F1), (F2) and (F3) of Figure 4, then* $\texttt{Reach}(\texttt{CDS}) \subseteq \texttt{Safe}$.

THEOREM 12. *[Relative Completeness] Let* $\texttt{CDS} := (X, \texttt{Init}, f)$ *be a CDS and* `Safe` *be a closed set such that* $\texttt{Reach}(\texttt{CDS}) \subseteq \texttt{Safe}$. *Let* $p \geq 0$ *be an inductive invariant such that* $p \geq 0 \Rightarrow \texttt{Safe}$. *Then, the following claims are true.*
(1) If $p = 0 \Rightarrow \vec{\nabla} p \neq 0$, *then* $p \geq 0$ *satisfies Conditions (C1), (C2), (C2') and (C3).*
(2) If p *is quadratic, then* $p \geq 0$ *satisfies Conditions (D1), (D2), (D2') and (D3).*
(3) If $p \geq 0$ *is* `convex`, *then* $p \geq 0$ *satisfies Conditions (F1), (F2) and (F3).*

Theorem 12 shows that the rules in Figure 3 are complete for a large class of practically useful invariants, namely, linear, quadratic, and convex invariants. Note that for a polynomial CDS and a semi-algebraic safe set, given a p, the inference rules in Figure 3 are formulas in the first-order theory of the reals, which is decidable [19]. It appears to be extremely difficult to come up with a simple and effective rule that is sound and complete for the class of *all* invariants of the form $p \geq 0$.

Example 7 *The set* $-(-x^2 - y^2 + 2y)^2 \geq 0$, *which geometrically is the circumference of a circle, is not* `convex`. *In fact, inference rules in Figure 3 and Figure 4 will all fail to prove that this set is an inductive invariant under the dynamics given by* $\frac{dx}{dt} = 1 - y$, $\frac{dy}{dt} = x$.

Discussion The rule in Figure 4 is related to the earlier rules via the observation that $p(\vec{u}+\vec{x})_1(f(\vec{u}))$ is equal to $L_f(p)(\vec{u})$. In the special case when $\vec{\nabla}p \neq 0$, the role of the witness direction g (in Figure 4) can be performed by $\vec{\nabla}p$. Thus, Figure 4 is also relatively complete for "smooth" sets and hence it is more powerful than the rules in Figure 3.

The rules above are complete for larger classes that what have been identified above. For example, the rule in Figure 3(Right) is complete for all p such that $p(\vec{x}) = 0 \wedge \vec{\nabla}(p)(\vec{x}) = 0 \Rightarrow (p(\vec{x}+\vec{y})_2(f(\vec{x})) \neq 0$, but we do not explore those results here.

Since Condition (F2) is based on Nagumo's criterion, which holds more generally, we can now easily generalize Condition (F2) from $p \geq 0$ to more general boolean combinations of polynomial inequalities. Let $\mathtt{In}(p, \vec{x}, f)$ be a predicate that denotes Condition (F2) applied to polynomial p at point $\vec{x}$ with vector field f. When the candidate invariant is $p_1 \geq 0 \ \wedge \ p_2 \geq 0$, Condition (F2) generalizes to $(p_1(\vec{x}) = 0 \wedge p_2(\vec{x}) > 0 \Rightarrow \mathtt{In}(p_1, \vec{x}, f)) \wedge (p_1(\vec{x}) > 0 \wedge p_2(\vec{x}) = 0 \Rightarrow \mathtt{In}(p_2, \vec{x}, f)) \wedge (p_1(\vec{x}) = 0 \wedge p_2(\vec{x}) = 0 \Rightarrow \mathtt{In}(p_1, \vec{x}, f) \wedge \mathtt{In}(p_2, \vec{x}, f))$. Similarly, when the candidate invariant is $p_1 \geq 0 \ \vee \ p_2 \geq 0$, then Condition (F2) generalizes to $(p_1(\vec{x}) = 0 \wedge p_2(\vec{x}) < 0 \Rightarrow \mathtt{In}(p_1, \vec{x}, f)) \wedge (p_1(\vec{x}) < 0 \wedge p_2(\vec{x}) = 0 \Rightarrow \mathtt{In}(p_2, \vec{x}, f)) \wedge (p_1(\vec{x}) = 0 \wedge p_2(\vec{x}) = 0 \Rightarrow \mathtt{In}(p_1, \vec{x}, f) \vee \mathtt{In}(p_2, \vec{x}, f))$.

Hybrid Systems Since hybrid systems extend CDSs with discrete transitions, and since the rule to handle discrete transitions is standard, the sound inference rules for hybrid systems can be obtained by combining the rule for continuous systems with the rule for discrete transitions. However, when using the rule for continuous systems, we can use any rule whose soundness is proved using Condition $(\mathcal{A}2)$ (such as rule in Figure 1(Right)), but we cannot use a rule whose soundness is proved using Condition (S2) (such as rule in Figure 4). The reason is that, as mentioned in Section 3, Condition (T2) is locally sound, whereas Condition (S2) is locally unsound, but only globally sound. In hybrid systems, due to the possibility of the presence of discrete transitions from the boundary, we need a sound condition that can verify invariance locally at every point.

Example 8 *We build a hybrid system to exploit the difference illustrated in Example 1. Consider a hybrid system that has only one mode, with dynamics* $f((x,y)) = (1,0)$ *and a discrete transition given by,* $x := -x$ *whenever* $x^2 + (y-1)^2 = 1 \wedge x > 0$. *Suppose initially,* $x^2 + (y-1)^2 \leq 1$ *and we want to show that this initial set is also an inductive invariant. We note that this set is not an invariant because there are trajectories leaving the invariant set from points* $(0,1)$ *and* $(0,0)$. *But Condition (S2) holds at both these points, and it also holds on all boundary points* from where there is no discrete transition. *The invariant set is inductive with respect to the discrete transitions. This shows that one has to be careful when generalizing rules based on Condition (S2) to hybrid systems.*

5 Conclusions

We presented several inference rules for safety verification of continuous systems and analyzed their soundness and relative completeness. We have a finite and sound rule that is also complete for the class of invariants containing convex and certain smooth semi-algebraic sets. It remains a challenge to discover an effectively checkable and sound rule that is complete for all semi-algebraic invariants.

References

[1] A. Abate, A. Tiwari, and S. Sastry. Box invariance for biologically-inspired dynamical systems. In *Proc. IEEE Conf. on Decision and Control, CDC*, pages 359–364, 2007.

[2] D. Beyer, T. Henzinger, R. Majumdar, and A. Rybalchenko. Invariant synthesis for combined theories. In *VMCAI*, volume 4349 of *LNCS*, pages 378–394, 2007.

[3] F. Blanchini. Set invariance in control. *Automatica*, 35:1747–1767, 1999.

[4] K. Burns and M. Gidea. *Differential Geometry and Topology: With a view to dynamical systems*. Chapman & Hall, 2005.

[5] P. Cuijpers and M. Reniers. Lost in translation: Hybrid-time flows vs real-time transitions. In *Proc. 11th HSCC*, volume 4981 of *LNCS*, pages 116–129. Springer, 2008.

[6] R. W. Floyd. Assigning meaning to programs. In *Proc. Symp. in Appl. Math*, 1967.

[7] S. Gulwani, S. Srivastava, and R. Venkatesan. Program analysis as constraint solving. In *Proc. ACM Conf. on Prgm. Lang. Desgn. and Impl. PLDI*, pages 281–292, 2008.

[8] S. Gulwani and A. Tiwari. Constraint-based approach for analysis of hybrid systems. In *Proc. 20th CAV*, volume 5123 of *LNCS*, pages 190–203. Springer, 2008.

[9] T. A. Henzinger. A theory of hyrid automata. In *Proc. 11th IEEE Logic in Comp. Sci. LICS*, pages 278–292, 1996.

[10] C. A. R. Hoare. An axiomatic basis of computer programming. *Comm. ACM*, 12(10):576–580, 1969.

[11] Deepak Kapur. Automatically generating loop invariants using quantifier elimination. In *Deduction and Applications*, 2005.

[12] R. M. Keller. Formal verification of parallel programs. *Comm. of the ACM*, 19(7), 1976.

[13] A. Platzer. Differential dynamic logic for hybrid systems. *J. Autom. Reasoning*, 41(2):143–189, 2008.

[14] A. Platzer and E. M. Clarke. Computing differential invariants of hybrid systems as fixedpoints. In *CAV*, volume 5123 of *LNCS*, pages 176–189. Springer, 2008.

[15] S. Prajna and A. Jadbabaie. Safety verification of hybrid systems using barrier certificates. In *HSCC*, volume 2993 of *LNCS*, pages 477–492, 2004.

[16] S. Prajna, A. Jadbabaie, and G. Pappas. A framework for worst-case and stochastic safety verification using barrier certificates. *IEEE Trans Aut Control*, 52(8), 2007.

[17] S. Sankaranarayanan, H. Sipma, and Z. Manna. Constructing invariants for hybrid systems. In *HSCC*, volume 2993 of *LNCS*, pages 539–554, 2004.

[18] S. Sankaranarayanan, H. Sipma, and Z. Manna. Non-linear loop invariant generation using gröbner bases. In *POPL*, pages 318–329, 2004.

[19] A. Tarski. *A Decision Method for Elementary Algebra and Geometry*. University of California Press, 1948.

[20] A. Tiwari. Generating box invariants. In *Proc. HSCC*, LNCS 4981. Springer, 2008.

[21] A. J. van der Schaft and J. M. Schumacher, editors. *An introduction to hybrid dynamical systems*, volume 251 of *Lecture Notes in Ctrl. and Inf. Sci.* Springer, 2000.