Semi-Algebraic Constant Reset Hybrid Automata - SACoRe

Alberto Casagrande*†, Carla Piazza†, Bud Mishra‡§
*PARADES, Via S.Pantaleo, 66, 00186 Roma, Italy
†DIMI, Università di Udine, Via delle Scienze, 206, 33100 Udine, Italy
‡Courant Institute of Mathematical Science, NYU, New York, U.S.A.
§NYU School of Medicine, 550 First Avenue, New York, 10016 U.S.A.

***Abstract*— In this paper we introduce and study a special class of hybrid automata, *Semi-Algebraic Constant Reset hybrid automata* (SACoRe). SACoRe automata are an extension of O-minimal automata over the reals in the case of flows obtained from non-autonomous systems of differential inclusions. Even though SACoRe automata do not have the finite bisimulation property, they do admit decision procedures for reachability and model checking for a limited fragment of CTL, by combining Tarski's decidability result over the reals and Michael's selection theorem.**

I. Introduction

The notion of *Hybrid Automata* was first introduced in [1] as a model and specification language for hybrid systems, i.e., systems consisting of a discrete program within a continuously changing environment. Since their introduction they have been widely used for the automatic verification of both natural and engineered systems.

In this paper we introduce and study a special class of such automata, *Semi-Algebraic Constant Reset hybrid automata* (SACoRe), whose characterizing conditions are based upon first-order theory over $(\mathbb{R}, 0, 1, +, *, =, <)$. In particular, a hybrid automaton of dimension k can be defined using only formulæ over k dimensional vectors of reals. The dynamics are defined through formulæ which can be obtained as solutions of non-autonomous systems of differential inclusions. The reset conditions have to be constant as in the case of O-minimal hybrid automata [2]. Even though SACoRe automata do not have the finite bisimulation property, the conditions we impose on their dynamics allow us to combine Tarski's result [3] and Michael's selection theorem [4] to translate reachability problems into first-order satisfiability problems over the reals.

The approach of exploiting Tarski's result and quantifier elimination to study hybrid automata has begun to be widely investigated in the last few years. For instance, Jirstrand [5] demonstrated, in the context of non-linear control system design, the use of `Qepcad` for the problems of computing reachability, stationarizable sets, range of controllable output, and curve-following. Subsequently, Anai [6] and Franzle [7] independently suggested the use of quantifier elimination for the verification of polynomial hybrid systems. Franzle went on to prove that progress, safety, state recurrence and reachability are semi-decidable using quantifier elimination [8] and developed "proof engines" for bounded model checking [9]. More recently, Lafferiere et al. [10] have again described a method based upon quantifier elimination for symbolic reachability computation of linear vector fields.

Lately, Ratschan and She [11] have suggested a new constraint propagation based abstraction refinement for the safety verification of hybrid systems with autonomous differential equations. Other recent developments include Becker et al.'s integration of bounded model checking and inductive verification [12]. Lanotte and Tini [13] have recently proved that the semi-algebraic hybrid automaton obtained by approximating each formula in any hybrid system definition with its Taylor polynomial is an over-approximation.

The novelty of our approach mainly lies in the use of continuous selection results [14] which allow us to consider non-autonomous differential inclusions. Moreover, as a direct consequence of continuous selection results, we can derive first-order formulæ to encode reachability problems with low structural complexity.

The paper is organized as follows. In Section II we introduce the syntax and the semantics of SACoRe automata. In Section III we show how to decide reachability. The fact that SACoRe automata do not have the finite bisimulation property is proved in Section IV. Section V is devoted to model checking of a fragment of CTL for SACoRe automata. Finally, Section VI ends the paper with some considerations about the use of SACoRe automata to study stability and robustness of biological systems. All the missing proofs can be found in [15].

II. SACoRe Hybrid Automata

A. Syntax

First, we introduce some notations and conventions. Capital letters Z_m, Z'_m, and Z^n_m, where $n, m \in \mathbb{N}$ denote variables ranging over $\mathbb{R}$. Analogously, Z denotes the vector of variables $\langle Z_1, \ldots, Z_k\rangle$; Z' denotes the vector $\langle Z'_1, \ldots, Z'_k\rangle$; and Z^n denotes the vector $\langle Z^n_1, \ldots, Z^n_k\rangle$. The temporal variables T and T' model time and range over $\mathbb{R}^+$. We use the small letters $p, q, r, s, \ldots$ to denote k-dimensional vectors of real numbers.

Occasionally, we will use the notation $\varphi[X_1, \ldots, X_m]$ to stress the fact that the set of free variables of the first-order formula φ may be included in the set of variables $\{X_1, \ldots,$

This work has been partially supported by the European Community Project IST-2001-33520 CC (Control and Computation), NSF's ITR programs, DARPA's BioCOMP/Biospice program, and NYU CCPR/DHS prgram.

$X_m\}$. By extension, if $\{X^1, \ldots, X^n\}$ is a set of variable vectors, $\varphi[X^1, \ldots, X^n]$ indicates that the free variables of φ are included in the set of components of $X^1, \ldots, X^n$. Moreover, given a formula $\varphi[X^1, \ldots, X^i, \ldots, X^n]$ and a vector p of the same dimension as the variable vector X^i, the formula obtained by component-wise substitution of X^i with p is denoted by $\varphi[X^1, \ldots, X^{i-1}, p, X^{i+1}, \ldots, X^n]$. If in φ the only free variables were the components of X^i, after the substitution we can compute the truth value of $\varphi[p]$.

We are now ready to formally introduce hybrid automata. For each state of a discrete automaton we have an invariant condition and a dynamic law. This dynamic law may depend on the initial conditions, i.e., on the values of the continuous variables at the beginning of the evolution in the state. The jumps from one discrete state to another are regulated by the activation and reset conditions.

Definition 1 (Hybrid Automata): A *hybrid automaton* $H = (Z, Z', \mathcal{V}, \mathcal{E}, Inv, Dyn, Act, Reset)$ of dimension k consists of the following components:

1) $Z = \langle Z_1, \ldots, Z_k\rangle$ and $Z' = \langle Z'_1, \ldots, Z'_k\rangle$ are two vectors of variables ranging over the reals $\mathbb{R}$;
2) $\langle \mathcal{V}, \mathcal{E}\rangle$ is a finite directed graph; the vertexes, $\mathcal{V}$, are called *locations*, or *control modes*, the directed edges, $\mathcal{E}$, are called *edges*, or *control switches*;
3) Each vertex $v \in \mathcal{V}$ is labeled by the two formulæ $Inv(v)[Z]$ and $Dyn(v)[Z, Z', T]$ such that if $Inv(v)[p]$ is true then $Dyn(v)[p, p, 0]$ is true; $InvSet = \{Inv(v)[Z] \mid v \in \mathcal{V}\}$ and $DynSet = \{Dyn(v)[Z, Z', T] \mid v \in \mathcal{V}\}$;
4) Each directed edge $e \in \mathcal{E}$ is labeled by the two formulæ$Act(e)[Z]$ and $Reset(e)[Z, Z']$; $ActSet = \{Act(e)[Z] \mid e \in \mathcal{E}\}$ and $ResetSet = \{Reset(e)[Z, Z'] \mid e \in \mathcal{E}\}$.

In our definitions, instead of the classical approach of using differential equations to define the flow, we use the formulæ in $DynSet$ to describe the continuous evolution without using derivatives. Our approach is similar to that followed in [16]. For instance, in [2], even though the automata are defined with differential equations, it is necessary to compute their solutions in order that the bisimulation algorithm can be applied, and express these solutions by $Dyn(v)[Z, Z', T]$, whose intuitive meaning is that from Z after T instants the continuous flow can reach Z'. Thus, our hybrid automata generalize several recently discovered notions in the hybrid systems theory. Note, as an example, that *O-minimal* hybrid automata [2], [16] are a subclass of our hybrid automata, since we do not impose restrictions on the formulæ and on the resets. Moreover, we admit an infinite number of flows, which can also be self-intersecting. Similarly, *Rectangular* hybrid automata [17] can be easily mapped into a subclass of our definition. In general, we are able to express all the hybrid automata defined using differential expressions, provided that either exact or approximated solutions of the differential expressions can be characterized with a formula.

Example 1: Consider this system of differential equations:

$$\begin{cases} \dot{Z}_1 &= 2Z_1 \\ \dot{Z}_2 &= Z_2 + 3 \end{cases}$$

Its solutions with initial conditions $Z_1(0) = z_1$ and $Z_2(0) = z_2$ are

$$\begin{cases} Z_1(t) &= z_1 e^{2t} \\ Z_2(t) &= (z_2 + 3)e^t - 3 \end{cases}$$

Translated in our notation this system corresponds to the following hybrid automaton $H = (Z, Z', \mathcal{V}, \mathcal{E}, Inv, Dyn, Act, Reset)$ where $Z = \langle Z_1, Z_2\rangle$ and $Z' = \langle Z'_1, Z'_2\rangle$ are variables over $\mathbb{R}^2$; $\mathcal{V} = \{v\}$ and $\mathcal{E} = \emptyset$; $Inv(v)[Z] \equiv \mathtt{true}$; $Dyn(v)[Z, Z', T] \equiv (Z'_1 = Z_1 e^{2T} \wedge Z'_2 = (Z_2 + 3)e^T - 3)$.

Hence, starting from the point $p_0 = \langle 1, 1\rangle$ we reach at time $T = 1$ the point $p_1 = \langle e^2, 4e - 3\rangle$ and at time $T = 2$ the point $p_2 = \langle e^4, 4e^2 - 3\rangle$. Notice that if we start from the point p_1 at time $T = 1$ we reach p_2, as we are using an autonomous system of differential equations.

Consider next the following system:

$$\begin{cases} \dot{Z}_1 &= 2t \\ \dot{Z}_2 &= 1 \end{cases}$$

We can express this in our notation with the hybrid automaton H' in which $\mathcal{V}$, $\mathcal{E}$, $InvSet$ are as in H, while $Dyn(v)[Z, Z', T]$ is $(Z'_1 = T^2 + Z_1 \wedge Z'_2 = T + Z_2)$.

Starting from the point $q_0 = \langle 1, 1\rangle$, we can reach at time $T = 1$ the point $q_1 = \langle 2, 2\rangle$ and at time $T = 2$ the point $q_2 = \langle 5, 3\rangle$. Notice that in this case if we start at time 0 from q_1 at time $T = 2$ we reach the point $q_3 = \langle 6, 3\rangle$ which cannot be reached starting from q_0. In fact, as this example illustrates, when the system of differential equations is not autonomous and the trajectories are not "transitive", the trajectories cannot be split and recombined.

As is well known, the afore-introduced hybrid automata are "undecidable", i.e., many of the classical problems regarding hybrid automata, such as *reachability* and *temporal logic model checking*, remain recalcitrant to a decision procedure [18] even when specialized to the kind of automata described above. Many subclasses of hybrid automata have been explored in the literature with the hope of proving decidability results under appropriate restrictions, e.g., O-minimal hybrid automata [2] and Rectangular hybrid automata [17] are two such well-known examples. In the rest of the paper, we will focus on decidability results for a new subclass of hybrid automata, we introduce here.

Following the approach of O-minimal hybrid automata, we require that the formulæ defining the invariants, the dynamics, the activations, and the resets be taken from an o-minimal theory. In particular, we focus on the first-order theory over the reals, as it suffices for all our areas of applications. Nonetheless, our results can be also applied to O-minimal extension of the reals, *mutatis mutandis*.

Definition 2 (Semi-Algebraic Automata): We call a hybrid automaton H *semi-algebraic* if the formulæ in $InvSet$, $DynSet$, $ActSet$, and $ResetSet$ are first-order formulæ over the reals, i.e., over $(\mathbb{R}, 0, 1, +, *, =, <)$.

In order to define this new class of automata, we also need to characterize the time instants, at which the automata, starting from a point p in a location v, can reach a point q, while remaining inside the invariant set of v. Such a characterization is possible when the automaton is semi-algebraic. We recall that an interval over $\mathbb{R}^+$ is a set of the form $\{r \in \mathbb{R}^+ \mid a \prec_1 r \prec_2 b\}$, where $\prec_1, \prec_2$ are in $\{<, \leq\}$, $a \in \mathbb{R}^+$, $b \in \mathbb{R}^+ \cup \{+\infty\}$, and $a \leq b$.

Lemma 1: Let H be a semi-algebraic hybrid automaton. Let $p \in \mathbb{R}^k$ be such that $Inv(v)[p]$ holds. The set of time instants T, satisfying the formula $\exists Z'(Dyn(v)[p, Z', T] \wedge Inv(v)[Z'])$, can be expressed as the union of a finite number of disjoint intervals of $\mathbb{R}^+$. One of these intervals contains the time instant 0.

The above lemma allows us to focus on the interval I_p^v of time instants, for which there are dynamics that start from p and remain inside the invariant of v—these dynamics are main objects of our interest. We use $\wp(\mathbb{R}^k)$ to denote the set of subsets of $\mathbb{R}^k$.

Definition 3 (I_p^v and F_p^v): Let H be a semi-algebraic hybrid automaton. Let v be a location of H and p be such that $Inv(v)[p]$ holds. I_p^v is the interval of time instants satisfying the following: $\forall T \in I_p^v\ \exists Z'(Dyn(v)[p, Z', T] \wedge Inv(v)[Z'])$; $0 \in I_p^v$, and I_p^v is maximal with respect to the first two requirements.

Define the function $F_p^v : I_p^v \to \wp(\mathbb{R}^k)$ as:

$$F_p^v(T) = \{q \mid Dyn(v)[p, q, T] \text{ and } Inv(v)[q]\}.$$

We will need to impose on the functions F_p^v some continuity conditions—in particular, we require *lower semi-continuity*, as defined below. For a complete treatment of this notion, please refer to [14].

Definition 4 (Lower semi-continuous function): Let $I \subseteq \mathbb{R}^k$ be an interval and $F\colon I \to \wp(\mathbb{R}^k)$. We define F to be *lower semi-continuous* (abbreviated, l.s.c.) if for each $t \in I$, for each $y \in F(t)$, and for each neighborhood U_y of y, there exists a neighborhood U_t of t (in I) such that for each $t' \in U_t$ it holds $F(t') \cap U_y \neq \emptyset$.

We now possess all the ingredients to introduce our class of hybrid automata.

Definition 5 (Semi-Algebraic Constant Reset Automata): We say that a hybrid automaton H is a *semi-algebraic constant reset* hybrid automaton, or simply a *SACoRe*, if:

1) H is semi-algebraic;
2) For each $v \in \mathcal{V}$, $p \in \mathbb{R}^k$ such that $Inv(v)[p]$ holds, the function F_p^v is lower semi-continuous, and for each $t \in I_p^v$ the set $F_p^v(t)$ is closed and convex;
3) Each formula $Reset(e)[Z, Z']$ is of the form $Reset(e)[Z']$, i.e., it does not depend on Z.

A SACoRe hybrid automaton is defined using first-order formulæ over the reals, and thus, exploits Tarski's results over the reals [3] to get decidability procedures. The condition 2 imposes a certain kind of continuity on the set of trajectories. Moreover, it requires that for each point p and for each time instant t the set of points reachable from p at time t is a closed convex set. This condition will allow us to exploit Michael's selection theorem [4] to find trajectories. The condition 3 is exactly the condition imposed on O-minimal hybrid automata.

Example 2: Let $H = (Z, Z', \mathcal{V}, \mathcal{E}, Inv, Dyn, Act, Reset)$ where $Z = \langle Z_1, Z_2\rangle$ and $Z' = \langle Z_1', Z_2'\rangle$; $\mathcal{V} = \{v\}$ and $\mathcal{E} = \{e\}$, where e goes form v to v; $Inv(v)[Z] \equiv (0 \leq Z_1 \leq 1 \wedge 0 \leq Z_2 \leq 1)$; $Dyn(v)[Z, Z', T] \equiv (Z_1' = T + Z_1 \wedge Z_2' \geq T^2 + Z_2)$; $Act(e)[Z] \equiv (Z_1 = 1 \vee Z_2 = 1)$; $Reset(e)[Z, Z'] \equiv (Z_1' = 1 \wedge Z_2' = 1)$.

The formulæ in H are first-order formulæ over the reals. If $p = \langle p_1, p_2\rangle$, with $0 \leq p_1, p_2 \leq 1$, then the function F_p^v is defined as $F_p^v(t) = \{\langle q_1, q_2\rangle \mid q_1 = t + p_1, q_2 \geq t^2 + p_2, \text{ and } 0 \leq q_1, q_2 \leq 1\}$. It is easy to see that $p \in F_p^v(0)$ and for each t the set $F_p^v(t)$ is closed and convex, since it is a segment. Moreover, this function is lower semi-continuous over the interval I_p^v. Finally, $Reset(e)[Z, Z']$ does not depend on Z. Hence, H is a SACoRe automaton.

O-minimals hybrid automata are easily seen as special cases of SACoRe automata. Since, in O-minimals hybrid automata, each point allows only one continuous algebraic flow from it, in this case, for each time instant t, the set $F_p^v(t)$ reduces to a singleton, which is obviously closed and convex. The continuity of the flow immediately implies the lower semi-continuity of $F_p^v(t)$ over I_p^v. On the other hand, the class SACoRe is not included in the class of O-minimals hybrid automata, since from each point we allow a set of flows. Moreover, our flows are not necessarily solutions of autonomous differential inclusions.

B. Semantics

Let H be a hybrid automaton of dimension k. The semantics of H is presented in terms continuous and discrete transitions as defined below.

Definition 6 (Hybrid Automata - Transitions): A *state* ℓ of H is a pair $\langle v, r\rangle$, where $v \in \mathcal{V}$ is a location and $r = \langle r_1, \ldots, r_k\rangle \in \mathbb{R}^k$ is an assignment of values for the variables of Z. A state $\langle v, r\rangle$ is said to be *admissible* if $Inv(v)[r]$ is true.

The *continuous reachability transition relation* $\to_C$ between admissible states is defined as follows:
$\langle v, r\rangle \to_C \langle v, s\rangle$ iff
there exists $f : \mathbb{R}^+ \to \mathbb{R}^k$ continuous function such that $r = f(0)$, there exists $t \geq 0$ such that $s = f(t)$, and for each $t' \in [0, t]$ the formulæ $Inv(v)[f(t')]$ and $Dyn(v)[r, f(t'), t']$ are true.

The *discrete reachability transition relation* $\to_D$ between admissible states is defined as follows:
$\langle v, r\rangle \to_D \langle u, s\rangle$ iff
It holds $\langle v, u\rangle \in \mathcal{E}$ and the formulæ $Act(\langle v, u\rangle)[r]$ and $Reset(\langle v, u\rangle)[s]$ are true.

Building upon continuous and discrete transitions, we can introduce notions of *trace* and *reachability*. A trace is a sequence of continuous and discrete transitions. A point s is reachable from a point r if there is a trace starting from r and ending in s. We use the notation $\ell \to \ell'$ to denote that either $\ell \to_C \ell'$ or $\ell \to_D \ell'$.

Definition 7 (Hybrid Automata - Reachability): Let I be either $\mathbb{N}$ or an initial finite interval of $\mathbb{N}$. A *trace* of H is a sequence $\ell_0, \ell_1, \ldots, \ell_i$ with $i \in I$, also denoted by $(\ell_i)_{i \in I}$, of admissible states such that:

- For each $i \in I$, $i > 0$, it holds $\ell_{i-1} \to \ell_i$;
- If $\ell_i \to_C \ell_{i+1}$, then $\ell_{i+1} \not\to_C \ell_{i+2}$.

A point $r \in \mathbb{R}^k$ *reaches* a point $s \in \mathbb{R}^k$ if there exists a trace $\ell_0, \ldots, \ell_n$ of H such that $\ell_0 = \langle v, r \rangle$ and $\ell_n = \langle u, s \rangle$, for some $v, u \in \mathcal{V}$.

We use $ReachSet(r)$ to denote the set of points reachable from r. Moreover, given a region $R \subseteq \mathbb{R}^k$ we use $ReachSet(R)$ to denote the set $\cup_{r \in R} ReachSet(r)$.

We impose the condition that, in a trace, continuous transitions do not occur consecutively. If we only consider automata whose flows are solutions of autonomous differential inclusions, there the continuous transition relation is transitive, and all their traces, containing sequence of consecutive continuous transitions, can be reduced to a trace without such consecutive continuous transitions. In general, it may be the case that the continuous transition relation is not transitive (see H' in Example 1). In this case, if we start from a point r in a location v, as long as we remain inside v, it is reasonable to consider only those points reachable from r, which satisfy the dynamics conditions imposed on r, i.e. $Dyn(v)[r, Z', T]$. Similarly we allow that a point r may reach a point s passing through a point u, while s may not be reachable from u. Such apparently paradoxical situation can occur when the dynamics are solutions of non-autonomous differential inclusions, since in this case the evolution from a point depends on time instant, at which the point is reached.

Usually two properties hold in temporal semantics [19], [20]: the *suffix closure* which requires that the suffix of a trace is a trace; the *fusion closure* which requires that the concatenation of two traces is a trace. Our semantics is suffix closed. On the other hand, fusion closure holds in our semantics only for the following case: when two traces tr_1 and tr_2 are concatenated such that either tr_1 ends with a discrete transition or tr_2 starts with a discrete transition. In fact, fusion closure implies that the system behavior depends only on the current state, and not on the past. This property does not always hold in our dynamics, since we allow solutions of non-autonomous differential inclusions.

We recall that given a finite directed graph G a *path* of G is a sequence $v_0, v_1, \ldots, v_n, \ldots$ of nodes of G such that for each $i \geq 0$ there exists an edge of G connecting v_i to v_{i+1}. Given a trace of H we can identify a path of $\langle \mathcal{V}, \mathcal{E} \rangle$ as follows.

Definition 8 (Corresponding Path): Let H be a SACoRe automaton. Let $tr = \langle v_0, r_0 \rangle, \ldots, \langle v_n, r_n \rangle$ be a trace of H. The *corresponding path* of tr is the path $ph = v'_0, \ldots, v'_m$ of the graph $\langle \mathcal{V}, \mathcal{E} \rangle$ obtained by considering the discrete transitions occurring in tr. In this case, we also say that ph *corresponds* to tr.

Notice that for each trace tr there exists always a unique path ph which corresponds to tr.

C. Reachability and Model Checking

Given a SACoRe hybrid automaton H and a starting region $R \subseteq \mathbb{R}^k$ characterized by a first-order formula ρ over the reals, we may wish to compute the region $ReachSet(R) \subseteq \mathbb{R}^k$ of points that can be reached starting from a point in R and following a trace of H.

More generally, given a formula Q of a temporal logic, we may also be interested in determining the points of R which satisfy Q. Let us introduce here the syntax and semantics of $\text{CTL}_{-\mathtt{X}}$, CTL without the next operator (see [21]).

Definition 9 ($CTL_{-\mathtt{X}}$ - Syntax): Let $\mathcal{P}$ be a set of *propositional symbols* and $P \in \mathcal{P}$. The formulæ of CTL over $\mathcal{P}$ are defined by the following grammar:

$$Q ::= \quad P \mid Q_1 \vee Q_2 \mid \neg Q_1 \mid \mathtt{E}(Q_1 \,\mathtt{U}\, Q_2) \mid \mathtt{A}(Q_1 \,\mathtt{U}\, Q_2) \mid \mathtt{EF}\, Q_1 \mid \mathtt{AF}\, Q_1 \mid \mathtt{EG}\, Q_1 \mid \mathtt{AG}\, Q_1$$

We avoid using the next operator, since it requires the introduction of a temporized semantics (see, e.g., [22]), thus taking us out of the scope of this paper.

In the case of O-minimals hybrid automata, reachability as well as other temporal logic proprieties are checked through bisimulation (see [2]) as follows: first, a finite discrete automaton A bisimilar to the hybrid automaton H is computed; next, the property is checked on A. Since bisimulation strongly preserves both reachability and temporal formulæ, the results obtained on A are correct, by definition. This technique can be applied whenever we consider a class $\mathcal{C}$ of hybrid automata, which has the finite bisimulation property, i.e., each automaton in $\mathcal{C}$ has a finite bisimulation quotient. Unfortunately, the class of SACoRe does not possess the finite bisimulation property, as we will show in Section IV.

Our approach will instead exploit both Tarski's decidability result [3] for first-order formulæ over ($\mathbb{R}$, 0, 1, $+$, $*$, $=$, $<$) and Michael's selection theorem for set-valued maps. More specifically, Michael's selection theorem will guarantee the correctness of a translation into appropriate first-order formulæ of our reachability and model checking problems, whereas Tarski's result will provide us the decidability.

III. Reachability

In this section, we demonstrate how the reachability problem over SACoRe automata can be reduced to a first-order satisfiability problem. We start characterizing the sets I^v_p.

Lemma 2: Let H be a SACoRe automaton. Consider the first-order formula

$$Tp(v)[Z, T] \stackrel{\text{def}}{=} \forall 0 \leq T' \leq T \exists Z' (Dyn(v)[Z, Z', T'] \wedge Inv(v)[Z']).$$

Assume r to be such that $Inv(v)[r]$ holds. It follows that:

$$t \in I^v_r \text{ iff } Tp(v)[r, t] \text{ is true.}$$

Theorem 1: Let H be a SACoRe automaton, satisfying the first-order formula below:

$$\begin{aligned} Reach(v)[Z,Z'] &\stackrel{\text{def}}{=} Inv(v)[Z] \wedge Inv(v)[Z'] \\ &\exists T \geq 0 (Dyn(v)[Z,Z',T] \wedge Tp(v)[Z,T]). \end{aligned}$$

Then following holds:

$$\langle v, r\rangle \rightarrow_C \langle v, s\rangle \text{ iff } Reach(v)[r,s] \text{ is true.}$$

Proof: ($\Rightarrow$) By Definition 6 we have that: $\langle v,r\rangle \rightarrow_C \langle v,s\rangle$ iff there exists $f : \mathbb{R}^+ \rightarrow \mathbb{R}^k$ continuous function such that $r = f(0)$, there exists $t \geq 0$ such that $s = f(t)$, and for each $t' \in [0,t]$ the formulæ $Inv(v)[f(t')]$ and $Dyn(v)[r, f(t'), t']$ are true.

From the fact that for each $t' \in [0,t]$ $Dyn(v)[r, f(t'), t'] \wedge Inv(v)[f(t')]$ is true, we deduce that $Tp(v)[r,t]$ is true. Hence we see that $Inv(v)[r]$, $Inv(v)[s]$, $Dyn(v)[r,s,t]$, and $Tp(v)[r,t]$ are all true, as stated.

($\Leftarrow$) By hypothesis, $Inv(v)[r]$ and $Inv(v)[s]$ are true and thus, there exists $t \in \mathbb{R}^+$ such that $Dyn(v)[r,s,t] \wedge Tp(v)[r,t]$ holds. Hence by Lemma 2 we have that $t \in I_r^v$. Moreover, s belongs to $F_r^v(t)$, which is lower semi-continuous with convex and closed images. Consider the function $\tilde{F} : [0,t] \rightarrow \wp(\mathbb{R}^k)$ defined as:

$$\tilde{F}(T) = \begin{cases} \{r\} & \text{if } T = 0 \\ F_p^v(T) & \text{if } 0 < T < t \\ \{s\} & \text{if } T = t \end{cases}$$

It is immediately seen that for each t' in $[0,t]$ $\tilde{F}(t')$ is closed and convex. We prove that $\tilde{F}$ is lower semi-continuous on $[0,t]$. Let $t' \in [0,t]$. We need to consider three distinct cases: (a) $t' = 0$; (b) $0 < t' < t$; (c) $t' = t$.

(a) If $t' = 0$ and $y \in \tilde{F}(0)$, then $y = r$. Let U_r be a neighborhood of r. Since, F_r^v is lower semi-continuous there exists a neighborhood U_0 of 0 in I_r^v such that for each t'' in U_0 it holds that $F_r^v(t'') \cap U_r \neq \emptyset$. Since, $[0,t] \subseteq I_r^v$ we get that $U_0' = U_0 \cap [0,t)$ is a neighborhood of 0 in $[0,t]$. If $t'' \in U_0'$, there are two possible subcases: either $t'' = 0$ or $0 < t'' < t$. If $t'' = 0$, then $\tilde{F}(0) \cap U_r = \{r\} \neq \emptyset$. If, on the other hand, $0 < t'' < t$, then $\tilde{F}(t'') \cap U_r = F_r^v(t'') \cap U_r \neq \emptyset$.

(b) If $0 < t' < t$ and $y \in \tilde{F}(t')$, then $y \in F_r^v(t')$. Let U_y be a neighborhood of y. Since F_r^v is lower semi-continuous, there exists a neighborhood $U_{t'}$ of t' in I_r^v such that for each t'' in $U_{t'}$ it holds that $F_r^v(t'') \cap U_y \neq \emptyset$. Since $t' \in (0,t) \subseteq I_r^v$, we conclude that $U_{t'}' = U_{t'} \cap (0,t)$ is a neighborhood of t' in $[0,t]$. If $t'' \in U_{t'}'$, then $\tilde{F}(t'') \cap U_r = F_r^v(t'') \cap U_r \neq \emptyset$.

(c) If $t' = t$ and $y \in \tilde{F}(t)$, then $y = s$. Let U_s be a neighborhood of s. Since F_r^v is lower semi-continuous, there exists a neighborhood U_t of t in I_r^v such that for each t'' in U_t, it holds that $F_r^v(t'') \cap U_s \neq \emptyset$. Since $[0,t] \subseteq I_r^v$, we get that $U_t' = U_t \cap (0,t]$ is a neighborhood of t in $[0,t]$. If $t'' \in U_t'$, then there are two possible subcases: namely, either $t'' = t$ or $0 < t'' < t$. If $t'' = t$, then $\tilde{F}(0) \cap U_s = \{s\} \neq \emptyset$. If $0 < t'' < t$, then $\tilde{F}(t'') \cap U_s = F_r^v(t'') \cap U_s \neq \emptyset$.

Since $\tilde{F} : [0,t] \rightarrow \wp(\mathbb{R}^k)$ is lower semi-continuous, for each t' in $[0,t]$, $\tilde{F}(t')$ is closed and convex; and since $[0,t]$ is a metric space, and $\mathbb{R}^k$ is a Banach space, by Michael's selection theorem [4] we may deduce the following: there exists $f : [0,t] \rightarrow \mathbb{R}^k$ continuous selection from $\tilde{F}$. Hence, by definition of continuous selection (see [14]), f is a continuous function such that for each $t' \in [0,t]$ it holds $f(t') \in \tilde{F}(t')$. From this last statement, we further deduce that: $f(0) = r$; $f(t) = s$; for each $0 < t' < t$ it holds that $f(t') \in F_r^v(t')$, i.e., $Dyn(v)[r, f(t'), t']$ and $Inv(v)[f(t')]$. In particular, consider the function $\tilde{f} : \mathbb{R}^+ \rightarrow \mathbb{R}^k$ defined as:

$$\tilde{f}(T) = \begin{cases} f(T) & \text{if } T \in [0,t] \\ s & \text{if } T > t \end{cases}$$

We have demonstrated that $\tilde{f}$ satisfies all the hypothesis required to conclude that $\langle v, r\rangle \rightarrow_C \langle v, s\rangle$, as desired. ∎

One may observe that for any edge $\langle v, u\rangle \in \mathcal{E}$ the discrete reachability is characterized by the first-order formula

$$Reach(\langle v,u\rangle)[Z,Z'] \stackrel{\text{def}}{=} Act(\langle v,u\rangle)[Z] \wedge Reset(\langle v,u\rangle)[Z'].$$

Given a point $r \in \mathbb{R}^k$, we see that the first-order formula $Reach(v)[r, Z']$, as defined in Theorem 1, and with free variables in Z', characterizes the set of points reachable from r in the node v using only continuous dynamics. Similarly, the first-order formula $Reach(e)[r, Z']$ defines the set of points reachable from r using the discrete transition e.

Now suppose that a point r reaches a point s through a trace tr, whose corresponding path is $ph = v, u$. Since, by Definition 1, $Dyn(v)[r,r,0]$ and $Dyn(u)[s,s,0]$ hold, we see that $\langle v, r\rangle \rightarrow_C \langle v, r\rangle$ and $\langle u, s\rangle \rightarrow_C \langle u, s\rangle$. Hence, tr is equivalent to tr' of the form $\langle v, r\rangle \rightarrow_C \langle v, r_1\rangle \rightarrow_D \langle u, s_1\rangle \rightarrow_C \langle u, s\rangle$. Thus, the reachability can always be expressed through a trace whose corresponding path is ph and results in the following first-order formula:

$$\begin{aligned} &Reach(v,u)[Z, Z^1, Z^2, Z'] \stackrel{\text{def}}{=} \\ &\quad Reach(v)[Z, Z^1] \wedge Reach(\langle v,u\rangle)[Z^1, Z^2] \\ &\quad \wedge Reach(u)[Z^2, Z']. \end{aligned}$$

If we have a path $ph = v_0, v_1, \ldots, v_h$ in the graph $\langle \mathcal{V}, \mathcal{E}\rangle$, then following two cases are possible: either it corresponds to a trace of H or it does not. In both cases, we can express the desired reachability relation with a first-order formula, which characterizes all the pairs of $\mathbb{R}^k$ that can be connected in H through a trace corresponding to path ph:

$$\begin{aligned} &Reach(ph)[Z, Z^1, \ldots, Z^{2h}, Z'] \stackrel{\text{def}}{=} \\ &\quad Reach(v_0)[Z, Z^1] \wedge Reach(\langle v_0, v_1\rangle)[Z^1, Z^2] \wedge \ldots \\ &\quad \wedge Reach(v_h)[Z^{2h}, Z']. \end{aligned}$$

In $Reach(ph)[Z, Z^1, \ldots, Z^{2h}, Z']$, we have $2h$ free variables, and no quantifiers. The following lemma proves that $Reach(ph)[Z, Z^1, \ldots, Z^{2h}, Z']$ is correct and complete.

Lemma 3: Let H be a SACoRe automaton, let $ph = v_0, v_1, \ldots, v_h$ be a path in $\langle \mathcal{V}, \mathcal{E}\rangle$. It holds that r reaches s through a trace tr whose corresponding path is ph iff $Reach(ph)[r, Z^1, \ldots, Z^{2h}, s]$ is satisfiable.

Hence, r reaches s if and only if there exists a path ph of $\langle \mathcal{V}, \mathcal{E}\rangle$ and has a formula $Reach(ph)[Z, Z^1, \ldots, Z^{2h}, Z']$ as a witness to this fact. So, if we just considered the disjunction of all the formulæ for all the paths of $\langle \mathcal{V}, \mathcal{E}\rangle$, we would characterize reachability. Unfortunately, if $\langle \mathcal{V}, \mathcal{E}\rangle$ has a cycle, then it has an infinite number of paths. However, we can exploit the fact that SACoRe have constant resets and ignore all the paths of $\langle \mathcal{V}, \mathcal{E}\rangle$ whose length exceeds $|\mathcal{E}|$.

Definition 10: Let H be a SACoRe automaton. Let P be the set of paths of $\langle \mathcal{V}, \mathcal{E}\rangle$ of length at most $m = |\mathcal{E}|$. Define the first-order formula $\mathcal{R}[Z, Z^1, \ldots, Z^{2m}, Z']$ as follows:

$$\mathcal{R}[Z, Z^1, \ldots, Z^{2m}, Z'] \stackrel{\text{def}}{=} \bigvee_{ph \in P} \mathit{Reach}(ph)[Z, Z^1, \ldots, Z^{2m}, Z'].$$

Theorem 2: Let H be a SACoRe automaton. It holds that $s \in \mathit{ReachSet}(r)$ iff $\mathcal{R}[r, Z^1, \ldots, Z^{2m}, s]$ is satisfiable.

We can now characterize the set of points reachable from a first-order definable set $R \subseteq \mathbb{R}^k$.

Corollary 1: Let $R \subseteq \mathbb{R}^k$ be the set of points which satisfies the first-order formula $\rho[Z]$. The set $\mathit{ReachSet}(R)$ is characterized by the first-order formula

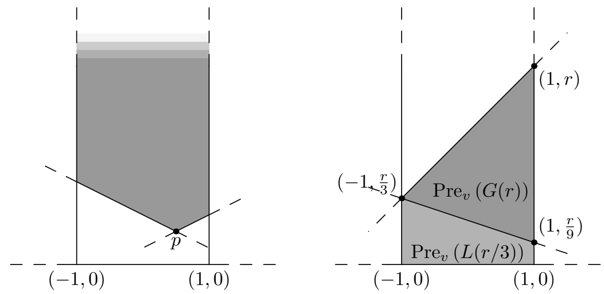
$$\mathcal{R}(R)[Z'] \stackrel{\text{def}}{=} \exists Z(\rho[Z] \wedge \exists Z^1, \ldots, Z^{2m}\, \mathcal{R}[Z, Z^1, \ldots, Z^{2m}, Z']).$$

Thus we have reduced our reachability problem to that of deciding the satisfiability of an existential semi-algebraic formula involving $v = O((|\mathcal{V}|+|\mathcal{E}|)k)+N(\rho))$ variables in total degree $d = \max[\deg(\mathit{Inv}), \deg(\mathit{Act}), \deg(\mathit{Dyn}), \deg(\rho)]$ and involving $s = O(|P| + |\rho|)$ polynomial equations, inequations and inequalities, where N and $\deg$ denote the number of variables and total degree, respectively used in the semi-algebraic description of Inv, Act, Dyn, ρ, etc. In addition, if we assume that the coefficients of the polynomials can be stored with at most L bits, then the total time complexity (bit-complexity) [23] of the decision procedure is $(L \log L \log\log L)(s/v)^v d^{O(v)}$. This exponential complexity has its origin in Collins' double-exponential complexity algorithm and its relatives, all to some degree based upon a cylindrical algebraic decomposition algorithm [24]. Later Hoon Hong, using many useful and practical heuristics, created the first practical quantifier elimination software `Qepcad`. Alternative CAD-based methods have been proposed Grigoriev [25] and Renegar [26] that are doubly exponential in the number of quantifier alternations rather than the number of variables. New quantifier elimination approaches have been proposed by Basu [27]. More importantly, symbolic algebraic geometry holds many other powerful tools such as Groebner bases and characteristic sets in its arsenal, whose utility is just beginning to be examined.

IV. SACoRe and Bisimulation

In this section we prove that there exists a SACoRe which does not admit a finite bisimulation quotient. In particular we prove that the hybrid automaton $H_{\text{inf}} = (Z, Z', \mathcal{V}, \mathcal{E}, \mathit{Inv}, \mathit{Dyn}, \mathit{Act}, \mathit{Reset})$ where:

- $Z = \langle Z_1, Z_2\rangle$ and $Z' = \langle Z'_1, Z'_2\rangle$, where Z_1, Z_2, Z'_1 and Z'_2 are variables over $\mathbb{R}$,
- $\mathcal{V} = \{v\}$ and $\mathcal{E} = \{e\}$, where e goes form v to v,
- $\mathit{Inv}(v)[Z] \equiv (-1 \le Z_1 \le 1 \wedge Z_2 > 0)$,



(a) H_{inf}'s formula Dyn. (b) Preimages.

Fig. 1. Dynamic and preimage for the automaton H_{inf}.

- $\mathit{Dyn}(v)[Z, Z', t] \equiv \mathrm{up}\,[Z, Z'] \wedge \mathrm{up}'\,[Z, Z']$, where $\mathrm{up}\,[Z, Z'] \equiv Z'_2 \ge Z_2 Z'_1 + Z_2(1 - Z_1)$ and $\mathrm{up}'\,[Z, Z'] \equiv Z'_2 \ge -Z_2 Z'_1 + Z_2(1 + Z_1)$,
- $\mathit{Act}(e)[Z] \equiv (Z_1 = 1 \wedge 0 < Z_2 \le 1)$,
- $\mathit{Reset}(e)[Z, Z'] \equiv (Z'_1 = -1 \wedge 0 < Z'_2 \le 1)$,

is a SACoRe and does not admit a finite bisimulation.

Lemma 4: H_{inf} is a SACoRe automaton.

To prove that the automaton H_{inf} does not admit finite bisimulation, we have to introduce some set definitions. In particular, let $\mathcal{R}(e)$, $\mathcal{A}(e)$ and $\mathcal{I}(v)$ be subsets of $\mathbb{R}^2$ such that $\mathcal{R}(e) = \{p \mid \exists Z \in \mathbb{R}^2\ \mathit{Reset}(e)[Z, p]\}$, $\mathcal{A}(e) = \{p \mid \mathit{Act}(e)[p]\}$ and $\mathcal{I}(e) = \{p \mid \mathit{Inv}(v)[p]\}$, respectively. Furthermore, for all $P \subseteq \mathbb{R}^2$, for each $v \in \mathcal{V}$ and for each $\langle v, u\rangle \in \mathcal{E}$ let $\mathrm{Pre}_v\,(P) = \{p \mid q \in P\ \langle v, p\rangle \to_C \langle v, q\rangle\}$ and $\mathrm{Pre}_{\langle v,u\rangle}\,(P) = \{p \mid q \in P\ \langle v, p\rangle \to_D \langle u, q\rangle\}$, respectively. Since H_{inf} is a SACoRe automaton, it satisfies the Condition 3 ("constant reset condition") of the SACoRe automaton Definition 5 and then it follows that:

$$\mathrm{Pre}_e\,(P) = \begin{cases} \emptyset & \text{if } P \cap \mathcal{R}(e) = \emptyset \\ \mathcal{A}(e) & \text{if } P \cap \mathcal{R}(e) \neq \emptyset \end{cases}$$

Thus, as reported in [2], if H_{inf} admits a finite bisimulation then the Algorithm 1 terminates, when started with an initial partition: $\mathcal{S}_v = \{\mathcal{R}(e), \mathcal{A}(e), \mathcal{I}(v) \setminus (\mathcal{R}(e) \cup \mathcal{A}(e))\}$.

Algorithm 1 Bisimulation Algorithm for Hybrid System

for $v \in \mathcal{V}$ **do**
  while $\exists P, P' \in \mathcal{S}_v$ such that $\emptyset \neq P \cap \mathrm{Pre}_v\,(P') \neq P$ **do**
    $P_1 \leftarrow P \cap \mathrm{Pre}_v\,(P')$
    $P_2 \leftarrow P \setminus \mathrm{Pre}_v\,(P')$
    $\mathcal{S}_v \leftarrow (\mathcal{S}_v \setminus \{P\}) \cup \{P_1, P_2\}$
  end while
end for
$X/\sim \leftarrow \bigcup_v (v, \mathcal{S}_v)$

Following lemmas, however, allow us to conclude that the Algorithm 1 does not terminate on H_{inf} and consequently, H_{inf} does not admit finite bisimulation.

Lemma 5: Let $G(r)$ be the subset of $\mathbb{R}^2$ such that $G(r) = \{\langle p_1, p_2\rangle \mid p_1 = 1 \wedge 0 < p_2 \le r\}$. For the automaton H_{inf}, it holds that $\text{Pre}_v(G(r)) = \{p \mid 3p_2 \leq r(p_1+2) \wedge \mathit{Inv}(v)[p]\}$, where $p = \langle p_1, p_2\rangle$ and $v \in \mathcal{V}$.

Lemma 6: Let $L(r)$ be the subset of $\mathbb{R}^2$ such that $L(r) = \{\langle p_1,\, p_2\rangle \mid p_1 = -1 \wedge 0 < p_2 \leq r\}$. The automaton H_{inf} satisfies $\text{Pre}_v(L(r)) = \{p \mid 3p_2 \leq r(2-p_1) \wedge \mathit{Inv}(v)[p]\}$, where $p = \langle p_1, p_2\rangle$ and $v \in \mathcal{V}$.

Notice that, for the automaton H_{inf}, $L(1)$ and $G(1)$ are equal to $\mathcal{R}(e)$ and $\mathcal{A}(e)$, respectively.

Theorem 3: The automaton H_{inf} does not admit finite bisimulation.

Proof: Our proof that H_{inf} does not admit finite bisimulation relies on showing that the Algorithm 1 does not terminate on H_{inf}. At the start of the computation, the Algorithm 1 uses $\mathcal{S}_v = \{\mathcal{R}(e), \mathcal{A}(e), \mathcal{I}(v) \setminus (\mathcal{R}(e) \cup \mathcal{A}(e))\}$ as an initial partition. As $L(1) = \mathcal{R}(e)$ and $G(1) = \mathcal{A}(e)$, $\mathcal{S}_v$ can be written as $\mathcal{S}_v = \{L(1), G(1), \mathcal{I}(v) \setminus (L(1) \cup G(1))\}$. If $p = \langle p_1, p_2\rangle$ then, by Lemma 6 and G's definition:

$$\begin{aligned}\text{Pre}_v(L(r)) \cap G(r') &= \{Z \mid p_2 \leq \frac{r}{3}(2-p_1) \wedge \mathit{Inv}(v)[Z] \wedge \\ &\qquad p_1 = 1 \wedge 0 < p_2 \leq r'\} \\ &= \{Z \mid p_2 \leq \frac{r}{3} \wedge \mathit{Inv}(v)[Z] \wedge \\ &\qquad p_1 = 1 \wedge 0 < p_2 \leq r'\} \\ &= G\left(\frac{r}{3}\right).\end{aligned}$$

Similarly, by Lemma 5 and L's definition: $\text{Pre}_v(G(r')) \cap L(r) = L\left(\frac{r'}{3}\right)$. Thus, if $r < 3r'$ and $r, r' \in \mathbb{R}^+$ then $\emptyset \neq \text{Pre}_v(L(r)) \cap G(r') \neq G(r')$ and then the algorithm removes $G(r')$ from $\mathcal{S}_v$ and it inserts the sets $G\left(\frac{r}{3}\right)$ and $G(r') \setminus G\left(\frac{r}{3}\right)$ in $\mathcal{S}_v$. Otherwise, $r \geq 3r'$ holds and if $r, r' \in \mathbb{R}^+$ then $3r > r \geq 3r' > r'$. It follows that $\emptyset \neq \text{Pre}_v(G(r')) \cap L(r) \neq L(r)$ and then the algorithm removes $L(r)$ from $\mathcal{S}_v$ and it inserts the sets $L\left(\frac{r'}{3}\right)$ and $L(r) \setminus L\left(\frac{r'}{3}\right)$ in $\mathcal{S}_v$. Hence, since the initial partition contains both $L(1)$ and $G(1)$, during the subsequent computation steps, there will exist $r, r' \in (0,1]$ such that $L(r), G(r') \in \mathcal{S}_v$. Moreover at each computation steps $\exists P, P' \in \mathcal{S}_v \mid \emptyset \neq \text{Pre}_v(P) \cap P' \neq P'$— in particular, if $r < 3r'$ then $P = L(r)$ and $P' = G(r')$, since, Otherwise, $P = G(r')$ and $P' = L(r)$. It follows then that the Algorithm 1 does not terminate, leading to the conclusion that H_{inf} does not admit finite bisimulation. ■

The next corollary follows from Lemma 4 and Theorem 3.

Corollary 2: There exist SACoRe automata that do not admit finite bisimulation.

V. CTL Model Checking

Despite the absence of a bisimulation result for SACoRe, we can still show that a substantial and interesting fragment of $\text{CTL}_{-\texttt{X}}$ can be decided over SACoRe automata, building upon the decidability of reachability. Since this fragment, we will shortly introduce, is not included in LTL, it is not possible to use simulation equivalence to reduce the model.

Given a SACoRe automaton H of dimension k, we consider a set $\mathcal{P} = \{P_1[Z], \ldots, P_m[Z]\}$ of atomic propositions whose elements are first-order formulæ over the reals with k free-variables. The labeling functions associates to each proposition $P[Z]$ of $\mathcal{P}$ the set of states of H in which $P[Z]$ holds, i.e., $\mathit{Label}(P[Z]) = \{\langle v, r\rangle \mid P[r] \text{ holds}\}$.

Next, consider the set Ψ of formulæ defined by the following grammar.

$$Q ::= \quad P[Z] \mid \neg P[Z] \mid Q_1 \vee Q_2 \mid \texttt{EF}\, Q_1 \mid \texttt{AG}\, Q_1$$

Notice that the formula in $\texttt{EFAG}P[Z]$ which belongs to Ψ distinguishes models which are simulation equivalent (see [15]).

Given a SACoRe automaton H and a formula $Q \in \Psi$ we can decide $\langle v, r\rangle \models Q$ by reducing the problem to a first-order formula validity problem as follows.

Definition 11: Given $Q \in \Psi$, and a state v of H, let $Ph(v)$ be the set of paths of $\langle \mathcal{V}, \mathcal{E}\rangle$ starting from v of length at most $m = |\mathcal{E}|$. We define the formula $\mathcal{M}(Q, v)[Z]$ by induction on Q as follows:

- $\mathcal{M}(P[Z], v)[Z]$ is $\mathit{Inv}(v)[Z] \wedge P[Z]$;
- $\mathcal{M}(\neg P[Z], v)[Z]$ is $\mathit{Inv}(v)[Z] \wedge \neg P[Z]$;
- $\mathcal{M}(Q_1 \vee Q_2, v)[Z]$ is $\mathcal{M}(Q_1, v)[Z] \vee \mathcal{M}(Q_2, v)[Z]$;
- $\mathcal{M}(\texttt{EF}\, Q_1, v)[Z]$ is
$$\bigvee\nolimits_{ph \in Ph(v)} (\exists Z^* Z' (\mathit{Reach}(ph)[Z, Z^*, Z'] \wedge \mathcal{M}(Q_1, u_{ph})[Z']));$$
- $\mathcal{M}(\texttt{AG}\, Q_1, v)[Z]$ is
$$\bigwedge\nolimits_{ph \in Ph(v)} (\forall Z^* Z' (\mathit{Reach}(ph)[Z, Z^*, Z'] \rightarrow \mathcal{M}(Q_1, u_{ph})[Z']));$$

where we use Z^* for the sequence $Z^1, \ldots, Z^{2m}$, while for each $ph \in Ph(v)$ we use $u_{ph} \in \mathcal{V}$ for the last node of ph.

Since an existential formula $\texttt{EF}\, Q_1$ of Ψ requires only that Q_1 be true in one reachable point, whereas a universal formula $\texttt{AG}\, Q_1$ of Ψ requires that Q_1 be true at all reachable points, we convince ourselves that our translations into first order formulæ are correct.

Theorem 4: Let $Q \in \Psi$. It holds that:

$$\langle v, r\rangle \models Q \ \text{ iff } \ \mathcal{M}(Q, v)[r] \text{ is true.}$$

VI. Conclusions

Here, we have presented a new class of hybrid automata, and dubbed it SACoRe (Semi-Algebraic Constant Reset). They have many attractive properties, even though they do not have the finite bisimulation property. For instance, we discovered that reachability and a limited fragment of CTL are decidable over SACoRe automata. Our decidability results are novel as they exploit Tarski's decidability result over the reals [3] and Michael's selection theorem [4]. SACoRe automata properly extend O-minimal automata allowing non-autonomous differential inclusions instead of autonomous differential equations. We can easily extend our class of automata exploiting other selection theorems (see, e.g., [28]).

SACoRe automata provide a very general framework and yet allow one to verify properties in many fields of natural and engineered systems. In particular, they are useful when, as is often the case, lack of measurements for kinetic parameters of the underlying system of differential equations forces one to describe the flows, replacing the equations by differential inclusions. Many examples, illustrating the power of this approach, may be found in the study of stability and robustness of non-autonomous parametric systems. Instead of using simulations and punctual analysis, our method allows one to automatically analyze these properties by checking formulæ of the form $\mathtt{EF}\,\mathtt{AG}\,Q_1$ for an appropriate Q_1, whose choice depends on the system. Consider for instance the "multi-stage" regulation model for the bacterial chemotaxis network presented in [29]. This model formalizes several important protein-protein interactions in a bacterial biochemical pathway using a system of differential equations. Such a system appears in a parameterized form and its parameters represent the number of receptors of a particular substance, their specific activity and their influence on overall receptors activity. While in [29] the verification of the system adaptivity to changes of the receptor activity level is performed through simulation and punctual analysis, it would be possible to automate this verification using a SACoRe automaton. In particular, in this case, we can assume that from each initial condition $\langle \mathrm{CheYp}_0, \mathrm{CheAs\text{-}CheZ}_0, \mathrm{CheZ(active)}_0\rangle$ we can reach at time t all the points $\langle \mathrm{CheYp}_t, \mathrm{CheAs\text{-}CheZ}_t, \mathrm{CheZ(active)}_t\rangle$ such that CheYp_t, $\mathrm{CheAs\text{-}CheZ}_t$, and $\mathrm{CheZ(active)}_t$ are reachable from CheYp_0, $\mathrm{CheAs\text{-}CheZ}_0$, and $\mathrm{CheZ(active)}_0$, respectively, when the parameter A ranges between 0.3 and 0.5. In this way we introduce in the model two approximations: we assume that if $\mathrm{CheYp}_t^{0.3}$ is the value reached at time t with $A = 0.3$ and $\mathrm{CheYp}_t^{0.5}$ is the value reached at time t with $A = 0.5$, then all the values in $[\mathrm{CheYp}_t^{0.3}, \mathrm{CheYp}_t^{0.5}]$ are reachable at time t; we consider reachable at time t the cartesian product of the intervals reachable from each component. These approximations allow to satisfy the condition imposed on SACoRe automata. Moreover, we can still prove that CheYp always finally returns to its steady state. Notice that to perform our analysis we do not need to introduce the parameter A as a variable of the automaton. It is sufficient to introduce it as an existentially quantified variable in the dynamic definitions.

In the future we intend to deeply investigate the applications of SACoRe automata in the study of both natural and engineered systems. We also plan to analyze possible extensions with non-constant resets.

Acknowledgments. We thank prof. F. Zanolin and dott. P. Baiti for useful discussions.

References

[1] R. Alur, C. Courcoubetis, T. A. Henzinger, and P. H. Ho, "Hybrid Automata: An Algorithmic Approach to the Specification and Verification of Hybrid Systems," in *Hybrid Systems*, ser. LNCS, R. L. Grossman, A. Nerode, A. P. Ravn, and H. Richel, Eds. Springer, 1992, pp. 209–229.

[2] G. Lafferriere, G. J. Pappas, and S. Sastry, "O-minimal Hybrid Systems," *Mathematics of Control, Signals, and Systems*, vol. 13, pp. 1–21, 2000.

[3] A. Tarski, *A Decision Method for Elementary Algebra and Geometry*. Univ. California Press, 1951.

[4] E. Michael, "Continuous selections I," *Ann. of Math.*, vol. 63, pp. 361–382, 1956.

[5] M. Jirstrand, "Nonlinear Control System Design by Quantifier Elimination," *J. Symb. Comput.*, vol. 24, no. 2, pp. 137–152, 1997.

[6] H. Anai, "Algebraic Approach to Analysis of Discrete-Time Polynomial Systems," in *European Control Conference (ECC'99)*, 1999.

[7] F. Martin, "Analysis of Hybrid Systems: An ounce of realism can save an infinity of states," in *Computer Science Logic (CSL'99)*, ser. LNCS, J. Flum and M. Rodríguez-Artalejo, Eds., vol. 1683. Springer, 1999, pp. 126–140.

[8] M. Fränzle, "What Will Be Eventually True of Polynomial Hybrid Automata?" in *Theoretical Aspects of Computer Software (TACS'01)*, ser. LNCS, N. Kobayashi and B. C. Pierce, Eds., vol. 2215. Springer, 2001, pp. 340–359.

[9] M. Fränzle and C. Herde, "Efficient Proof Engines for Bounded Model Checking of Hybrid Systems," in *FMICS*, 2004.

[10] G. Lafferriere, G. J. Pappas, and S. Yovine, "Symbolic Reachability Computation for Families of Linear Vector Fields," *J. Symb. Comput.*, vol. 32, no. 3, pp. 231–253, 2001.

[11] S. Ratschan and Z. She, "Safety verification of hybrid systems by constraint propagation based abstraction refinement," in *Hybrid Systems: Computation and Control (HSCC'05)*, ser. LNCS, M. Morari and L. Thiele, Eds., vol. 3114. Springer, 2005, pp. 573–589.

[12] B. Becker, M. Behle, F. Eisenbrand, M. Fränzle, M. Herbstritt, C. Herde, J. Hoffmann, D. Kroening, B. Nebel, I. Polian, and R. Wimmer, "Bounded model checking and inductive verification of hybrid discrete-continuous systems," in *GI/ITG/GMM Workshop*, 2004.

[13] R. Lanotte and S.Tini, "Taylor Approximation for Hybrid Systems," in *Hybrid Systems: Computation and Control (HSCC'05)*, ser. LNCS, M. Morari and L. Thiele, Eds., vol. 3114. Springer, 2005, pp. 402–416.

[14] J. P. Aubin and A. Cellina, *Differential Inclusions*, ser. A Series of Comprehensive Studies in Mathematics. Springer, 1984, vol. 264.

[15] A. Casagrande, C. Piazza, and B. Mishra, "Semi-Algebraic Constant Reset Hybrid Automata - SACoRe," March 2005. [Online]. Available: http://fsv.dimi.uniud.it/papers/SACoRe

[16] T. Brihaye, C. Michaux, C. Rivière, and C. Troestler, "On O-Minimal Hybrid Systems," in *Hybrid Systems: Computation and Control (HSCC'04)*, ser. LNCS, R. Alur and G. J. Pappas, Eds., vol. 2993. Springer, 2004, pp. 219–233.

[17] T. A. Henzinger and P. W. Kopke, "State Equivalences for Rectangular Hybrid Automata," in *Proc. of Int. Conference on Concurrency Theory (Concur'96)*, ser. LNCS, U. Montanari and V. Sassone, Eds., vol. 1119. Springer, 1996, pp. 530–545.

[18] T. A. Henzinger, P. W. Kopke, A. Puri, and P. Varaiya, "What's decidable about hybrid automata?" in *Proc. of ACM Symposium on Theory of Computing (STOCS'95)*, 1995, pp. 373–382.

[19] E. A. Emerson and J. Y. Halpern, "Sometimes and Not Never revisited: on branching versus linear time temporal logic," *Journal of the ACM*, vol. 33, no. 1, pp. 151–178, 1986.

[20] K. Abrahamson, "Decidability and Expressiveness of Logics of Programs," Ph.D. dissertation, University of Washington at Seattle, 1980.

[21] M. C. Browne, E. M. Clarke, D. L. Dill, and B. Mishra, "Automatic verification of sequential circuits using temporal logic," *IEEE Transactions on Computers*, vol. 35, no. 12, pp. 1035–1044, 1986.

[22] R. Alur, C. Courcoubetis, and D. Dill, "Model-checking in dense real-time," *Information and Computation*, vol. 104, no. 1, pp. 2–34, 1993.

[23] B. Mishra, *Algorithmic Algebra*. Springer, 1993.

[24] G. E. Collins, "Quantifier Elimination for the Elementary Theory of Real Closed Fields by Cylindrical Algebraic Decomposition," in *Proceedings of the Second GI Conference on Automata Theory and Formal Languages*, ser. LNCS, vol. 33. Springer, 1975, pp. 134–183.

[25] D. Grigoriev, "Complexity of Deciding Tarski Algebra," *Journal of Symbolic Computation*, vol. 5, pp. 65–108, 1988.

[26] J. Renegar, "On the Computational Complexity and Geometry of the First-order Theory of the Reals, parts I-III," *Journal of Symbolic Computation*, vol. 13, pp. 255–352, 1992.

[27] S. Basu, "An Improved Algorithm for Quantifier Elimination Over Real Closed Fields," in *IEEE Symposium on Foundations of Computer Science (FOCS'97)*, 1997, pp. 56–65.

[28] A. Bressan and G. Colombo, "Selections and representations of multifunctions in paracompact spaces," *Studia Math.*, vol. 103, pp. 209–216, 1992.

[29] G. Almogy, L. Stone, and N. Ben-Tal, "Multi-Stage Regulation, a Key to Reliable Adaptive Biochemical Pathways," *Biophysical Journal*, vol. 81, pp. 3016–3028, 2001.