Polynomial Interrupt Timed Automata

Béatrice Bérard[1,4], Serge Haddad[2,4,5], Claudine Picaronny[2,4,5], Mohab Safey El Din[1,4,5], and Mathieu Sassolas[3]

[1] Sorbonne Université, Université P. & M. Curie, LIP6, UMR 7606, Paris, France
[2] École Normale Supérieure de Cachan, LSV, UMR 8643, INRIA, Cachan, France
[3] Université Paris-Est, LACL, Créteil, France
[4] CNRS
[5] Inria

Abstract. Interrupt Timed Automata (ITA) form a subclass of stopwatch automata where reachability and some variants of timed model checking are decidable even in presence of parameters. They are well suited to model and analyze real-time operating systems. Here we extend ITA with polynomial guards and updates, leading to the class of polynomial ITA (PolITA). We prove that reachability is decidable in 2EXPTIME on PolITA, using an adaptation of the *cylindrical decomposition* method for the first-order theory of reals. Compared to previous approaches, our procedure handles parameters and clocks in a unified way. We also obtain decidability for the model checking of a timed version of CTL and for reachability in several extensions of PolITA.

1 Introduction

Hybrid Automata. Hybrid systems [14] combine continuous evolution of variables according to flow functions (described by differential inclusions) in control nodes, and discrete jumps between these nodes, where the variables can be tested by guards and updated. This class of models is very expressive and all relevant verification questions (*e.g.* reachability) are undecidable. For the last twenty years, a large amount of research was devoted to identifying subclasses with decidable properties, by restricting the continuous dynamics and/or the discrete behavior of the systems. Among these classes lie the well known Timed Automata (TA) [3], where all variables are *clocks* evolving with rate 1 w.r.t. to global time, guards are comparisons of clocks with rational constants, and updates are resets. It is proved in [15] that reachability becomes undecidable when adding one stopwatch, *i.e.*, a clock whose rate is either 0 or 1 depending on the state, to timed automata. Decidability results were also obtained for larger classes (see [5,2,15,17,4]), usually by building from the associated transition system (with uncountable state space) a finite abstraction preserving a specific class of properties, like reachability or those expressed by temporal logic formulas. In all these abstractions, a state is a pair composed of a control node and a polyhedron of variable values [15,17].

Interrupt Timed Automata. The class of Interrupt Timed Automata (ITA), incomparable with TA, was introduced in [8,10] as another subclass of hybrid

automata with a (time-abstract) bisimulation providing a finite quotient, thus leading to decidability of reachability and some variants of timed model checking. In a basic n-dimensional ITA, control nodes are organized along n levels, with n stopwatches (also called clocks hereafter), one per level. At a given level, the associated clock is active, while clocks from lower levels are frozen and clocks from higher levels are irrelevant. Guards are linear constraints and the clocks can be updated by linear expressions (using only clocks from lower levels). The hierarchical structure of ITA makes them particularly well suited for modeling systems with interruptions, like real-time operating systems. ITA were extended with parameters in [9], while preserving decidability.

Contribution. We define the class PolITA, of polynomial ITA, where linear expressions on clocks are replaced by polynomials with rational coefficients both for guards and updates. For instance, a guard at level 2 with clock x_2 can be of the form $P_1(x_1)x_2^2 + P_2(x_1) \geq 0$, where P_1 and P_2 are polynomials with single variable x_1, the clock of level 1. Thus, guards are more expressive than in the whole class of linear hybrid automata and classical polyhedron-based abstractions [1,12] are not sufficient to deal with these constraints. Since linear constraints are not always sufficient for modeling purposes, such guards can be useful. In addition, such guards can simulate irrational (algebraic) constraints, a case that becomes undecidable in the setting of timed automata [19]. Similar polynomials of variables for programs were considered in [20], although in an untimed setting.

We establish that reachability is decidable in 2EXPTIME for PolITA by adapting the cylindrical decomposition [13,6] related to the first order theory of reals. Observe however that not any decision procedure would be appropriate for our goal. Indeed this decomposition produces a finite partition of the state space, which is the basis for the construction of a finite bisimulation quotient. The first order theory of reals has already been used in several works on hybrid automata [17,4] but it was restricted to the dynamical part, with discrete jumps that must reinitialize the variables. Our adaptation consists in an on-the-fly construction avoiding to build the whole decomposition.

The construction can also be adapted to model checking of a timed extension of CTL. From an expressiveness point of view, we show that (contrary to ITA) PolITA are incomparable with stopwatch automata (SWA). We also prove that the decidability result still holds with several extensions: adding auxiliary clocks and parameters, and enriching the possible updates. In particular, parametric ITA [9] can be seen as a subclass of PolITA, and the complexity of our reachability algorithm is better than [9] (2EXPSPACE).

Outline. We describe the model of polynomial ITA in Section 2, with an example and the presentation of the verification problems. In Section 3 we informally present the cylindrical decomposition and the decision procedures for PolITA. Then in section 4, we detail these constructions with a special focus on the data structures and algorithmic schemes. Finally, we discuss expressiveness, describe extensions and conclude in Section 5. All missing proofs and constructions can be found in [11].

2 Polynomial ITA

We denote respectively by $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$ and $\mathbb{R}$ the sets of natural numbers, integers, rational and real numbers, with $\mathbb{R}_{\geq 0}$ for the set of non negative real numbers. Let $X = \{x_1, \ldots, x_n\}$ be a finite set of n variables called clocks. We write $\mathbb{Q}[x_1, \ldots, x_n]$ for the set of polynomials with n variables and rational coefficients.

A *polynomial constraint* is a conjunction of constraints of the form $P \bowtie 0$ where $P \in \mathbb{Q}[x_1, \ldots, x_n]$ and $\bowtie \in \{<, \leq, =, \geq, >\}$, and we denote by $\mathcal{C}(X)$ the set of polynomial constraints. We also define $\mathcal{U}(X)$, the set of *polynomial updates* over X, by: $\mathcal{U}(X) = \{\wedge_{x \in X} x := P_x \mid \forall x \ P_x \in \mathbb{Q}[x_1, \ldots, x_n]\}$.

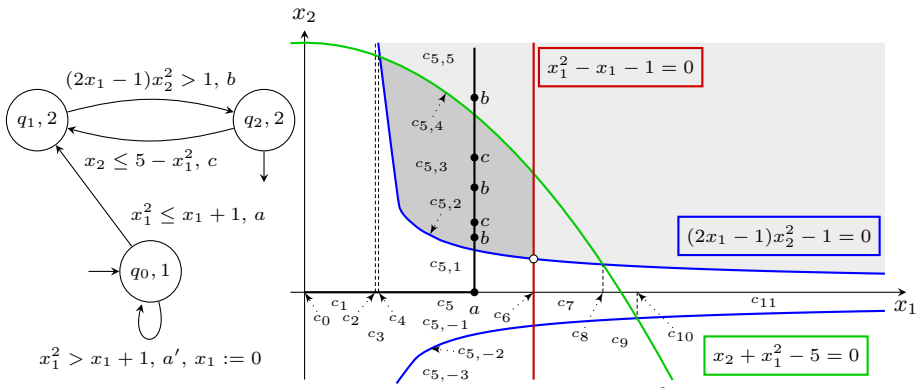
A valuation for X is a mapping $v \in \mathbb{R}^X$, also identified to the n-dimensional vector $(v(x_1), \ldots, v(x_n)) \in \mathbb{R}^n$. The valuation where $v(x) = 0$ for all $x \in X$ is denoted by $\mathbf{0}$. For $P \in \mathbb{Q}[x_1, \ldots, x_n]$ and v a valuation, the value of P at v is $P(v) = P(v(x_1), \ldots, v(x_n))$. A valuation v satisfies the constraint $P \bowtie 0$, written $v \models P \bowtie 0$, if $P(v) \bowtie 0$. The notation is extended to a polynomial constraint: $v \models \varphi$ with $\varphi = \bigwedge_i P_i \bowtie_i 0$ if $v \models P_i \bowtie_i 0$ for every i.

An update of valuation v by $u = \wedge_{x \in X} x := P_x$ in $\mathcal{U}(X)$ is the valuation $v[u]$ defined by $v[u](x) = P_x(v)$ for each $x \in X$. Hence an update is atomic in the sense that all variables are assigned simultaneously. For valuation v, delay $d \in \mathbb{R}_{\geq 0}$ and $k \in [1..n]$, the valuation $v' = v +_k d$, corresponding to *time elapsing of d for x_k*, is defined by $v'(x_k) = v(x_k) + d$ and $v'(x) = v(x)$ for $x \neq x_k$.

Definition 1 (PolITA). *A* polynomial interrupt timed automaton *(*PolITA*) is a tuple $\mathcal{A} = \langle \Sigma, Q, q_0, F, X, \lambda, \Delta \rangle$, where:*

- *Σ is a finite alphabet, with ε the empty word in Σ^*, the set of words over Σ;*
- *Q is a finite set of states, q_0 is the initial state, $F \subseteq Q$ is the set of final states;*
- *$X = \{x_1, \ldots, x_n\}$ consists of n interrupt clocks;*
- *the mapping $\lambda : Q \to \{1, \ldots, n\}$ associates with each state its level and $x_{\lambda(q)}$ is called the* active clock *in state q;*
- *$\Delta \subseteq Q \times \mathcal{C}(X) \times (\Sigma \cup \{\varepsilon\}) \times \mathcal{U}(X) \times Q$ is the set of transitions. Let $q \xrightarrow{\varphi, a, u} q'$ in Δ be a transition with $k = \lambda(q)$ and $k' = \lambda(q')$. The guard φ is a conjunction of constraints $P \bowtie 0$ with $P \in \mathbb{Q}[x_1, \ldots, x_k]$ (P is a polynomial over clocks from levels less than or equal to k). The update u is of the form $\wedge_{i=1}^{n} x_i := C_i$ with:*
  - *if $k > k'$, i.e. the transition decreases the level, then for $1 \leq i \leq k'$, $C_i = x_i$ and for $i > k'$, $C_i = 0$;*
  - *if $k \leq k'$ then for $1 \leq i < k$, $C_i = x_i$, $C_k = P$ for some $P \in \mathbb{Q}[x_1, \ldots, x_{k-1}]$ or $C_k = x_k$, and for $i > k$, $C_i = 0$.*

Example 1. PolITA $\mathcal{A}_0$ of Fig. 1a has alphabet $\{a, a', b, c\}$, two levels, with q_0 at level 1 and q_1, q_2 at level 2. The single final state is q_2. At level 1, only x_1 appears in guards and updates (here the only update is the reset of x_1 by action a'), while at level 2 guards use polynomials in both x_1 and x_2. In the sequel, the polynomials of $\mathcal{A}_0$ are denoted by $A = x_1^2 - x_1 - 1$, $B = (2x_1 - 1)x_2^2 - 1$ and $C = x_2 + x_1^2 - 5$.



(a) A sample PolITA $\mathcal{A}_0$. (b) Sample trajectory of $\mathcal{A}_0$ in $\mathbb{R}^2$. The axes are not orthonormal.

Fig. 1: A PolITA and an example of a trajectory.

A *configuration* (q, v) of $\mathcal{A}$ consists of a state q and a clock valuation v.

Definition 2. *The semantics of a* PolITA $\mathcal{A}$ *is defined by the (timed) transition system* $\mathcal{T}_\mathcal{A} = (S, s_0, \to)$*, where* $S = \{(q, v) \mid q \in Q,\ v \in \mathbb{R}^X\}$ *is the set of configurations, with initial configuration* $s_0 = (q_0, \mathbf{0})$*. The relation* $\to$ *on* S *consists of two types of steps:*
Time steps: *Only the active clock in a state can evolve, all other clocks are frozen. For a state* q *with active clock* $x_{\lambda(q)}$*, a time step of duration* $d \in \mathbb{R}_{\ge 0}$ *is defined by* $(q, v) \xrightarrow{d} (q, v')$ *with* $v' = v +_{\lambda(q)} d$*.*
Discrete steps: *There is a discrete step* $(q, v) \xrightarrow{a} (q', v')$ *if there exists a transition* $q \xrightarrow{\varphi, a, u} q'$ *in* Δ *such that* $v \models \varphi$ *and* $v' = v[u]$*.*

A run of a PolITA $\mathcal{A}$ is a path in the graph $\mathcal{T}_\mathcal{A}$ alternating time and discrete steps. For a given run ρ, the *trace* of ρ is the sequence of letters (or word) appearing in the path and the *timed word* of ρ is the sequence of letters along with the absolute time of the occurrence, *i.e.* the sum of all delays appearing before the letter. A run is accepting if it ends in a state of F. The *language* (resp. *timed language*) of $\mathcal{A}$ is the set of traces (resp. timed words) of accepting runs.

Example 2. In $\mathcal{A}_0$, the transition from q_0 to q_1 can only be fired before (or when) x_1 reaches $\frac{1+\sqrt{5}}{2}$, *i.e.* at the point labeled c_6 on Fig. 1b. Then, transition b from q_1 to q_2 can only be taken once x_2 reaches the grey areas. Transition c cannot be taken once the green curve has been crossed. Hence the loop bc can occur as long as the clock values remain in the dark gray area $c_{5,3}$, or on the green curve $c_{5,4}$. In the sequel, we show how to symbolically compute these sets, called *cells*. Since $q_2 \in F$, the run depicted in Fig. 1b is accepted by $\mathcal{A}$. The associated timed word (resp. trace) is $(a, 1.2)(b, 2.3)(c, 2.6)(b, 3.3)(c, 3.9)(b, 5.1)$ (resp. $abcbcb$).

Given a PolITA $\mathcal{A}$, the *reachability problem* asks, given a state q, whether there exists a valuation v and a path from $(q_0, \mathbf{0})$ to (q, v) in $\mathcal{T}_\mathcal{A}$.

The reachability procedure given in Section 3 relies on a finite abstraction of $\mathcal{T}_\mathcal{A}$. This abstraction needs to be refined enough to capture time elapsing, discrete jumps through the crossing of a transition, and keep constant the truth value of constraints $P \bowtie 0$. In the resulting model, a state will consist of an automaton state coupled with a *cell* of an appropriate *cylindrical decomposition*.

3 Cylindrical decomposition and reachability

3.1 Definition

The *cylindrical decomposition* is the basis of the first elementary decision procedure (more precisely 2EXPTIME) for the satisfiability of the first-order logic over reals [13][6]. A cylindrical decomposition of $\mathbb{R}^n$ consists of finite partitions of $\mathbb{R}, \mathbb{R}^2, \ldots, \mathbb{R}^n$ into *cells* such that the cells for $\mathbb{R}$ are open intervals or points and cells of $\mathbb{R}^{k+1}$ are obtained by lifting cells of $\mathbb{R}^k$ on the $k+1^{th}$ axis and then partitioning this axis with intervals and points in a "similar" way for all the points of the original cell.

Example 3. Fig. 1b partly depicts a cylindrical decomposition of $\mathbb{R}^2$. The cells of $\mathbb{R}_{\geq 0}$ are denoted by $c_0, \ldots, c_{11}$ (those of the negative part of the x_1 axis are not represented). The lifting of cell c_5 is $c_5 \times \mathbb{R}$ and is partitioned into cells $c_{5,-3}, c_{5,-2}, \ldots, c_{5,5}$. Given any $z \in c_5$, $\{z\} \times \mathbb{R}$ is partitioned in an open interval $c_{5,-3} \cap \{z\} \times \mathbb{R}$ followed by a point $c_{5,-2} \cap \{z\} \times \mathbb{R}$, etc. Observe that the mapping $z \mapsto c_{5,-2} \cap \{z\} \times \mathbb{R}$ is continuous.

Definition 3. *A cell of level k is a subset of $\mathbb{R}^k$ inductively defined as follows.*
- *When $k = 1$, it is either a point or an open interval.*
- *A cell C of level $k+1$ is based on a cell C' of level k. It has one of the following shapes.*
  1. *$C = \{(x, f(x)) \mid x \in C'\}$ with f a continuous function from C' to $\mathbb{R}$;*
  2. *$C = \{(x, y) \mid x \in C' \wedge l(x) < y < u(x)\}$ with $l < u$ continuous functions from C' to $\mathbb{R}$, possibly with $l = -\infty$ and/or $u = +\infty$.*

We are interested in a cylindrical decomposition *adapted* to finite families of polynomials $\mathcal{P} = \{\mathcal{P}_1, \ldots, \mathcal{P}_n\}$ with $\mathcal{P}_k \subseteq \mathbb{Q}[x_1, \ldots, x_k]$: in a cell of level k, the sign $(-, 0, +)$ of each polynomial in $\mathcal{P}_k$ is constant. Due to the definition of cells, a cylindrical decomposition is appropriately represented by a tree.

Definition 4. *A cylindrical decomposition of $\mathbb{R}^n$ adapted to $\mathcal{P} = \{\mathcal{P}_k\}_{k \leq n}$ such that $\mathcal{P}_k \subseteq \mathbb{Q}[x_1, \ldots, x_k]$, is a tree of cells inductively defined as follows:*
- *The root of the tree is the only cell of level 0, that is $\mathbb{R}^0$;*
- *Let C be a cell of level $k < n$ in the tree. There exists some $r \in \mathbb{N}$ and continuous functions f_i, for $1 \leq i \leq r$, with $-\infty = f_0 < f_1 < \ldots < f_r < f_{r+1} = +\infty$, such that the (ordered) children of C at level $k+1$ in the tree are the cells $C_0 = \{(x, y) \mid x \in C \wedge f_0(x) < y < f_1(x)\}$, $C_1 =$*

[6] Later on, an EXPSPACE procedure was proposed in [7].

$\{(x, f_1(x)) \mid x \in C\}$, $C_2 = \{(x, y) \mid x \in C \wedge f_1(x) < y < f_2(x)\}$, ... , $C_{2r} = \{(x, y) \mid x \in C \wedge f_r(x) < y < f_{r+1}(x)\}$.
For all $P \in \mathcal{P}_{k+1}$, *for all* $i \in \{0, \ldots, 2r\}$, *for all* $z, z' \in C_i$, $sign(P(z)) = sign(P(z'))$.

Example 4. For the PolITA of Fig. 1a, the relevant polynomials in $\mathbb{Q}[x_1]$ are those related to level 1: the clock x_1 itself and the polynomial $A = x_1^2 - x_1 - 1$ used in both guards from q_0, hence $\mathcal{P}_1 = \{x_1, A\}$. The relevant polynomials in $\mathbb{Q}[x_1, x_2]$ are those from level 2: x_2 and $B = (2x_1 - 1)x_2^2 - 1, C = x_2 + x_1^2 - 5$ associated with the guards from q_1 and q_2, so $\mathcal{P}_2 = \{x_2, B, C\}$. For the cells of level 1, c_4, c_8, c_{10} correspond to intersection points of graphs $B = 0$ and $C = 0$ projected on the x_1 axis, while c_2 corresponds to $\frac{1}{2}$, the root of the coefficient $2x_1 - 1$ of B. Other cells like c_1, c_3 correspond to intervals between roots. In cell $c_{5,3}$ of level 2, the guards of the transitions between q_1 and q_2 are satisfied.

The main elements for the effective construction of a cylindrical decomposition are given in Section 4. For the moment, we recall the result of [13]:

Theorem 1 ([13]). *For any family* $\mathcal{P} = \{\mathcal{P}_k\}_{k \leq n}$ *such that* $\mathcal{P}_k$ *is a finite subset of* $\mathbb{Q}[x_1, \ldots, x_k]$, *one can build a cylindrical decomposition of* $\mathbb{R}^n$ *adapted to* $\mathcal{P}$ *in 2EXPTIME, more precisely in* $(|\mathcal{P}| \cdot d)^{2^{O(n)}}$ *where* d *is the maximal degree of a polynomial of* $\mathcal{P}$.

3.2 Reachability for PolITA

We now use this decomposition to build a finite abstraction of the set of configurations of a PolITA, which leads to the decidability of the reachability problem.

Theorem 2. *Reachability for* PolITA *is decidable in time* $(d|\mathcal{A}|)^{2^{O(n)}}$ *where* n *is the number of clocks in* $\mathcal{A}$ *and* d *the maximal degree of polynomials appearing in* $\mathcal{A}$*; thus in polynomial time when the number of clocks is fixed.*

Let $\mathcal{A} = \langle \Sigma, Q, q_0, F, X, \lambda, \Delta \rangle$ be a PolITA with $X = \{x_1, \ldots, x_n\}$. We define $Poly(\mathcal{A})$ as the set of all polynomials appearing in guards and updates of $\mathcal{A}$ (including all clocks) as follows: P belongs to $Poly(\mathcal{A})$ iff (1) P is a clock, (2) P occurs in a guard $P \bowtie 0$, or (3) $P = x_i - P_i$ where $x_i := P_i$ is an update.

We denote by $\mathcal{D}_\mathcal{A}$ a cylindrical decomposition adapted to $Poly(\mathcal{A})$, with $\mathcal{D}^1_\mathcal{A}, \ldots, \mathcal{D}^n_\mathcal{A}$ for the set of cells at the respective levels $1, \ldots, n$ so that for $1 \leq k \leq n$, $\mathcal{D}^k_\mathcal{A}$ is a decomposition of $\mathbb{R}^{\{x_1, \ldots, x_k\}}$.

We define a finite transition system $\mathcal{R}_\mathcal{A}$ with states in $Q \times \mathcal{D}_\mathcal{A}$. The states can also be partitioned according to levels as $\bigcup_{k=1}^n \lambda^{-1}(k) \times \mathcal{D}^k_\mathcal{A}$. Indeed, given a configuration (q, v) with $\lambda(q) = k$, the clocks of level $i > k$ are irrelevant and so v can be identified as a point in $\mathbb{R}^{\{x_1, \ldots, x_k\}}$. We now define the transitions of $\mathcal{R}_\mathcal{A}$ as follows.

Time successors. Let $succ \notin \Sigma$ be a letter representing time elapsing. Let (q, C) be a state of $\mathcal{R}_\mathcal{A}$, with $\lambda(q) = k$, and let $\underline{C} \in \mathcal{D}^{k-1}_\mathcal{A}$ be the projection of C onto $\mathbb{R}^{k-1}$ and $-\infty = f_0 < \cdots < f_{r+1} = +\infty$ be the functions dividing $\underline{C}$ as in Definition 4. The *succ* transitions are defined as follows:

– if $C = \{(x, f_i(x)) \mid x \in \underline{C}\}$ for some $i \in \{1, \ldots, r\}$, then there is a transition $(q, C) \xrightarrow{succ} (q, C')$ where $C' = \{(x, y) \mid x \in \underline{C}, f_i(x) < y < f_{i+1}(x)\}$;
– if $C = \{(x, y) \mid x \in \underline{C}, f_{i-1}(x) < y < f_i(x)\}$ for some $i \in \{1, \ldots, r\}$, then there is a transition $(q, C) \xrightarrow{succ} (q, C')$ where $C' = \{(x, f_i(x)) \mid x \in \underline{C}\}$;
– otherwise, $C = \{(x, y) \mid x \in \underline{C}, f_r(x) < y < f_{r+1}(x)\}$, and there is a self-loop labeled by *succ*: $(q, C) \xrightarrow{succ} (q, C)$.

In all the above cases, C' is called the *time successor* of C (in the last case, C is its own time successor). Since the decomposition is *cylindrical*, time elapsing according to the current clock corresponds to moving to the "next" cell.

Proposition 1 (Correctness w.r.t. time elapsing). *Let v be a valuation belonging to a cell C of level k.*

– *There exists $d > 0$ such that the elapsing of d time units for x_k yields a valuation $v +_k d \in C'$, the time successor of C.*
– *For any $0 < d' < d$, the elapsing of d' time units for x_k yields a valuation $v +_k d$ that is either in C or in C'.*

Discrete successors. Since $\mathcal{D}_{\mathcal{A}}$ is adapted to $Poly(\mathcal{A})$ which contains all guards and updates we can write $C \models \varphi$ whenever $v \models \varphi$ for some $v \in C$ and $C[u]$ for the unique cell $C' \in \mathcal{D}^k_{\mathcal{A}}$ such that for any valuation $v \in C$, $v[u] \in C'$. Discrete transitions of $\mathcal{A}$ are translated as follows into $\mathcal{R}_{\mathcal{A}}$: if $(q, \varphi, a, u, q') \in \Delta$ and $C \models \varphi$, there is a transition $(q, C) \xrightarrow{a} (q', C[u])$. Since the decomposition provides *sign-invariant* cells with respect to the polynomials of $\mathcal{A}$, we have:

Proposition 2 (Correctness w.r.t. discrete steps).

– *If $(q, v) \xrightarrow{a} (q', v') \in \mathcal{T}_{\mathcal{A}}$, then $(q, C) \xrightarrow{a} (q', C') \in \mathcal{R}_{\mathcal{A}}$ with $v \in C$ and $v' \in C'$.*
– *If $(q, C) \xrightarrow{a} (q', C') \in \mathcal{R}_{\mathcal{A}}$ then for all $v \in C$ there exists $v' \in C'$ such that $(q, v) \xrightarrow{a} (q', v') \in \mathcal{T}_{\mathcal{A}}$.*

Since the number of cells in a cylindrical decomposition is doubly exponential in the number of clocks and polynomial in the number and maximal degree of polynomials to which it is adapted [6], we obtain the complexity stated in Theorem 2. By setting $\{(q, C) \mid q \in F\}$ as the set of final states of $\mathcal{R}_{\mathcal{A},\psi}$, this construction establishes that the untimed language of a PolITA is regular.

4 Effective construction and on-the-fly algorithm

4.1 Construction of a cylindrical decomposition

Building a cylindrical decomposition consists in two stages: the elimination stage that enlarges $\mathcal{P}$ and the lifting stage that builds the cylindrical decomposition using symbolic representations of sample points (one per cell).

Elimination stage. Starting from a cell C at level k, in order to get a partition at level $k+1$ adapted to $\mathcal{P}_{k+1}$, any two points $z, z' \in C$ should trigger a similar bevahiour for polynomials of $\mathcal{P}_{k+1}$, that we consider for our discussion as univariate polynomials of $\mathbb{Q}[x_1, \ldots, x_k][x_{k+1}]$ with variable x_{k+1}. More precisely, the properties we are looking for are:

– For all $P \in \mathcal{P}_{k+1}$ and for all z, z' in C, the number of real roots (counted with multiplicities) of the polynomials $P(z)$ and $P(z')$ in $\mathbb{R}[x_{k+1}]$ are equal (say μ_P). For $1 \leq i \leq \mu_P$ and $z \in C$, we denote by $r_{P,i}(z)$ the i^{th} real root of polynomial $P(z)$ (in increasing order) ;
– For all $P, Q \in \mathcal{P}_{k+1}$, for all $1 \leq i \leq \mu_P$ and $1 \leq j \leq \mu_Q$, for all z, z' in C, $r_{P,i}(z) \leq r_{Q,j}(z)$ implies $r_{P,i}(z') \leq r_{Q,j}(z')$.

These properties are analytical and do not provide insights on how to ensure them. Fortunately, it turns out that a simple effective sufficient condition exists: there is a finite subset of polynomials of $\mathbb{Q}[x_1, \ldots, x_k]$ denoted by $Elim_{x_{k+1}}(\mathcal{P}_{k+1})$ such that if z, z' satisfy $sign(R(z)) = sign(R(z'))$ for all $R \in Elim_{x_{k+1}}(\mathcal{P}_{k+1})$, then the above properties are satisfied.

To define $Elim_{x_{k+1}}(\mathcal{P}_{k+1})$, we need some notations. For $P = \sum_{i \leq p} a_i x_{k+1}^i$ with $a_i \in \mathbb{Q}[x_1, \ldots, x_k]$ for all i, $lcof(P)$ denotes the *leading coefficient* a_p. Since this leading coefficient is a polynomial and could be null for some $P(z)$, the set of truncations of P contains the possible "realizations" of P: $Tru(P) = \{\sum_{i \leq h} a_i x_{k+1}^i \mid \forall i > h \ a_i \notin \mathbb{R} \setminus \{0\} \wedge a_h \neq 0\}$. For instance, if $P = x_1 x_2^3 + (3x_1 + 1)x_2^2 + 5x_2 - 2$, then $Tru(P) = \{P, (3x_1 + 1)x_2^2 + 5x_2 - 2, 5x_2 - 2\}$. Given another polynomial, $Q = \sum_{i \leq q} b_i x_{k+1}^i \in \mathbb{Q}[x_1, \ldots, x_k][x_{k+1}]$, the *subresultants* $(sRes_i(P, Q))_{i \leq \max(p,q)}$ are polynomials of $\mathbb{Q}[x_1, \ldots, x_k]$ obtained as determinants of matrices whose items are coefficients of P and Q (see [6,11] for a formal definition of subresultants, a polynomial time computation and their properties).

Definition 5. *Let $\mathcal{P}_k$ be a finite subset of $\mathbb{Q}[x_1, \ldots, x_{k-1}][x_k]$ for $k > 1$. Then $Elim_{x_k}(\mathcal{P}_k)$ is the subset of $\mathbb{Q}[x_1, \ldots, x_{k-1}]$ defined for all $P, Q \in \mathcal{P}_k, R \in Tru(P)$, $T \in Tru(Q)$ by:*

– *If $lcof(R)$ does not belong to $\mathbb{Q}$ then $lcof(R) \in Elim_{x_k}(\mathcal{P}_k)$;*
– *If $deg(R) \geq 2$ then for all $sRes_j(R, \frac{\partial R}{\partial x_k})$ that are defined and do not belong to $\mathbb{Q}$, $sRes_j(R, \frac{\partial R}{\partial x_k}) \in Elim_{x_k}(\mathcal{P}_k)$;*
– *for all $sRes_j(R, T)$ that are defined and do not belong to $\mathbb{Q}$, $sRes_j(R, T) \in Elim_{x_k}(\mathcal{P})$.*

Using the properties of subresultants, one gets the following theorem whose implementation is the elimination stage of the cylindrical decomposition. Due to the quadratic blow up at each level of elimination the final number of polynomials is doubly exponential w.r.t. the original number.

Theorem 3. *Let $\mathcal{P} = \{\mathcal{P}_k\}_{k \leq n}$ be a family of finite set of polynomials such that $\mathcal{P}_k \subseteq \mathbb{Q}[x_1, \ldots, x_k]$. Define $\mathcal{Q}_n = \mathcal{P}_n$ and inductively $\mathcal{Q}_{k-1} = \mathcal{P}_{k-1} \cup Elim_{x_k}(\mathcal{Q}_k)$ for $k > 1$. Then there exists a cylindrical decomposition adapted to $\mathcal{Q}$ (and thus to $\mathcal{P}$).*

Example 5. Consider again the polynomials $B = (2x_1 - 1)x_2^2 - 1$ and $C = x_2 + x_1^2 - 5$ from the PolITA of Fig. 1a. Their subresultant of index 0 is $F = -2x_1^5 + x_1^4 + 20x_1^3 - 10x_1^2 - 50x_1 + 26$ which has precisely three real roots c_4, c_8, c_{10}: the x_1-coordinates of intersection points of graphs $B = 0$ and $C = 0$ mentioned previously.

Lifting stage. The starting point of the lifting stage is the family $\mathcal{P}$ appropriately enlarged by the elimination stage. In the cylindrical decomposition that we build, every cell C of level k is represented by a *sample point* inside the cell and the values of signs of all polynomials of set $\mathcal{P}_k$ on this point.

We consider representations of real subrings of the form $\mathbb{D} = \mathbb{Q}[\alpha_1, \ldots, \alpha_k]$ where the α_i's are algebraic numbers, *i.e.*, roots of polynomials in $\mathbb{Q}[x]$. Any real algebraic number α can be represented by a pair (n, P) where P is a non null polynomial in $\mathbb{Q}[x]$ such that $P(\alpha) = 0$ and n is the index of α in the ordered set of real roots of P. This representation is extended for real algebraic points $(\alpha_1, \ldots, \alpha_k)$ with the notion of *triangular systems*: α_1 is the n_1^{th} root of $P_1 \in \mathbb{Q}[x_1]$, α_2 is the n_2^{th} root of $P_2(\alpha_1)$ with $P_2 \in \mathbb{Q}[x_1][x_2]$, etc.

Definition 6 (Triangular system). *For $k \geq 1$, let $(\alpha_1, \ldots, \alpha_k)$ be a sequence of reals and let $\{(n_i, P_i)\}_{i=1}^k$ be such that for all i, n_i is a positive integer and $P_i \in \mathbb{Q}[x_1, \ldots, x_{i-1}][x_i]$. Then $\{(n_i, P_i)\}_{i=1}^k$ is a triangular system of level k for $(\alpha_1, \ldots, \alpha_k)$ if:*

- *P_1 is non null and α_1 is its n_1^{th} real root;*
- *For $1 \leq i < k$, $P_{i+1}(\alpha_1, \ldots, \alpha_i)$ is a non null polynomial of $\mathbb{Q}[\alpha_1, \ldots, \alpha_i][x_{i+1}]$ and α_{i+1} is its n_{i+1}^{th} real root.*

Example 6. Let us consider the point (α_1, α_2) depicted as a circle in Fig. 1b. This point is represented by the triangular system $((2, A), (2, B))$ where $A = x_1^2 - x_1 - 2$ and $B = (2x_1 - 1)x_2^2 - 1$. This means that α_1 is the 2^{nd} root of A and α_2 is the 2^{nd} root of $B(\alpha_1)$.

The interest of such a representation is its effectiveness: in a ring $\mathbb{D} = \mathbb{Q}[\alpha_1, \ldots, \alpha_k]$ associated with a triangular system one can compute (1) the sign of an item of $\mathbb{Q}[\alpha_1, \ldots, \alpha_k]$, (2) the number of real roots of $P(\alpha_1, \ldots, \alpha_k)$ with $P \in \mathbb{Q}[x_1, \ldots, x_k][x_{k+1}]$, (3) the sign realizations of a polynomial $Q(\alpha_1, \ldots, \alpha_k)$ on the real roots of a polynomial $P(\alpha_1, \ldots, \alpha_k)$, and one can order (with merge) the roots of $P(\alpha_1, \ldots, \alpha_k)$ and $Q(\alpha_1, \ldots, \alpha_k)$. All these procedures are performed in polynomial time (see for instance [11]).

The tree corresponding to the cylindrical decomposition is built top-down so that a triangular system is associated with a sample point of every cell and its sign realizations on the appropriate polynomials. Let us describe how, given a sample point $(\alpha_1, \ldots, \alpha_k)$, the partition over axis x_{k+1} can be built w.r.t. $\mathcal{P}_{k+1}$. First for all $P \in \mathcal{P}_{k+1}$, the number of roots of $P(\alpha_1, \ldots, \alpha_k)$ is determined. Then the roots of these polynomials are sorted and merged; their triangular system is the one associated with $(\alpha_1, \ldots, \alpha_k)$ extended by the polynomial for which they are roots. Then the open intervals between these roots or beyond these roots must be specified, to yield the *completed line partitioning*. Let (r, P) and (s, Q) be the borders of an open interval, then one selects as sample point, a root of $\frac{\partial(PQ)}{\partial x_{k+1}}$ located in the interval. Let (r, P) and $+\infty$ (resp. $-\infty$ and $(1, P)$) be the borders of the last (resp. first) open interval, then one selects $(r, P[x_{k+1} := x_{k+1} - 1])$ (resp. $(r, P[x_{k+1} := x_{k+1} + 1])$) as sample point. To achieve this step it remains to compute the sign realizations of $P(\alpha_1, \ldots, \alpha_k)$ for all $P \in \mathcal{P}_{k+1}$ on these sample points. Theorem 1 results from these two construction steps.

4.2 On-the-fly algorithm

The abstraction from Section 3 provides decidability of the reachability problem, by the algorithm that builds the finite graph $\mathcal{R}_{\mathcal{A}}$. However, building the complete graph is not efficient in practice, since it requires to build the set of all cells beforehand, even though usually most of them are unreachable. In the sequel, we show an on-the-fly construction of $\mathcal{R}_{\mathcal{A}}$ that reduces complexity in practice.

The key to the on-the-fly algorithm is to store only the part of the tree corresponding to the current sample point and its time successors. This construction relies on executing the lifting phase only when the level is increased and then only for the current sample point. As an illustration, in Fig. 1b, only the lifting for x_2 above c_5 has been represented, since it is the only relevant one with respect to the given trajectory. Note that liftings over sample points c_0 to c_6 have to be computed in order to build the reachable part of $\mathcal{R}_{\mathcal{A}_0}$. On the other hand, liftings over c_7 to c_{11} and over unrepresented cells to the left of c_0, need not, since level 2 is not reachable from these cells. As a result, we do not keep the whole tree but only part of it.

We show that this information is sufficient to compute the successors through time elapsing and transition firing. Although this pruning yields better performances in practice, the computational complexity in the worst case is not improved.

Definition 7 (Pruned tree). *Let $\{\mathcal{P}_k\}_{k\leq n}$ be the polynomials obtained by the elimination phase. The* pruned tree *for sample point $(\alpha_1, \ldots, \alpha_k)$ is the sequence of completed line partitionings for sample points $\{(\alpha_1, \ldots, \alpha_i)\}_{1\leq i\leq k}$. The pruned tree for the empty sample point ($k = 0$) is the line partitioning at level* 1.

A valuation $(v_1, \ldots, v_k, 0, \ldots, 0)$ at level k is represented by a sample point $(\alpha_1, \ldots, \alpha_k)$, or, equivalently, by a pruned tree for sample point $(\alpha_1, \ldots, \alpha_{k-1})$ and the index m of α_k in the line partitioning for $(\alpha_1, \ldots, \alpha_{k-1})$. In this representation, computing the time successors of $(\alpha_1, \ldots, \alpha_k)$ is simply done by incrementing m (if it is not the maximal index in the line partitioning).

The set of enabled discrete transitions can be generated by computing the signs of polynomials appearing in guards. When a discrete transition $q \xrightarrow{g,a,u} q'$ is chosen, there are three cases w.r.t. the level of states q and q'.

- The level decreases, *i.e.* $\lambda(q') < \lambda(q)$. Then the pruned tree corresponding to the new configuration is the truncation of the original pruned tree up to height $\lambda(q')$. Otherwise said, we "forget" line partitionings for levels above $\lambda(q')$; however, the partitionings are kept in memory to avoid redundant computations. The new index is the index of $\alpha_{\lambda(q')}$ in the partitioned line for this level.
- The level is unchanged, *i.e.* $\lambda(q') = \lambda(q) = k$. The only possible change of clock values is through an update $x_k := P$ with $P \in \mathbb{Q}[x_1, \ldots, x_{k-1}]$. The polynomial of degree 1 $R = x_k - P$ was added to $Poly(\mathcal{A})$ and its unique root α'_k appears in the line partitioning of level k. Note that in the triangular system representing $(\alpha_1, \ldots, \alpha'_k)$ it may appear as $((n_1, P_1), \ldots, (n_k, P_k))$ with $(n_k, P_k) \neq (1, R)$. Hence to determine the index in the partitioned line

the algorithm must actually determine the sign of R for all sample points of the line until 0 is found.
- The level increases, *i.e.* $\lambda(q') > \lambda(q)$. If there is an update of x_k, the same computations as above must be performed in order to find the new sample point corresponding to the valuation of clocks up to $\lambda(q)$. Then the pruned tree of height $\lambda(q')$ has to be computed (or retrieved). This is done by $\lambda(q') - \lambda(q)$ lifting steps. These lifting steps are applied on sample points of the form $(\alpha_1, \ldots, \alpha_{\lambda(q)}, 0, \ldots, 0)$, since all clocks are null for levels above $\lambda(q)$.

The on-the-fly algorithm builds the reachable part of $\mathcal{R}_\mathcal{A}$ as follows: the elimination phase is computed and the line for x_1 is partitioned. It starts with a queue containing q_0 with index corresponding to the root of x_1 (*i.e.* 0). Then until the queue is empty, it computes all (new) successors through time and discrete transitions, building the pruned tree as described above. As noted above, a line partitioning only needs to be computed once. In addition, and this also holds for the complete construction of $\mathcal{R}_\mathcal{A}$, the *triangular* structure of triangular systems enables a sharing of line partitioning at lower levels.

5 Conclusion and discussion

We extend ITA with polynomial expressions on clocks, and prove that reachability is decidable using the cylindrical decomposition. We also show that an on-the-fly construction of a class automaton is possible during the lifting phase of this decomposition.

We now mention several additional results proved in [11] but omitted here. The first one concerns the decidability of the model checking of $\mathsf{TCTL}_{\text{int}}$, a variant of TCTL [1], where only local clocks can be used in the formulas. The PolITA is equipped with atomic propositions that hold in states. Another direction was to investigate the expressive power of the model and try to extend it while keeping decidability of reachability. We first established that stopwatch automata and PolITA are incomparable. Then we proved that reachability is still decidable when including parameters in the expressions of guards and updates, with a better complexity than obtained in [9] (2EXPSPACE). We also extend the model by adding at each level i, a set of *auxiliary* clocks Y_i in addition to the *main* clock x_i. With several restrictions, we still obtain a decidability result for reachability. A last extension allows updates for clocks of levels lower than the current one. Again with some restrictions, decidability for reachability is preserved via a translation into a basic PolITA, similarly to [10] for ITA. Finally, as also presented in [10] for ITA, it is possible to extend the model of PolITA by adding timed automata at a lower level 0, producing a class that is stricly more expressive than timed automata.

An implementation is in progress to experiment the practical efficiency of the decision procedures. Since the construction still suffers from the doubly exponential complexity of the cylindrical decomposition, we plan to investigate if recent methods [16] with a lower complexity could be used to achieve reachability, possibly for a restricted version of PolITA. Another direction would be to enlarge the class of functions (like those studied in [18]) labelling guards and updates, still ensuring a finite bisimulation quotient.

References

1. Alur, R., Courcoubetis, C., Dill, D.L.: Model-checking in dense real-time. Information and Computation 104, 2–34 (1993)
2. Alur, R., Courcoubetis, C., Halbwachs, N., Henzinger, T.A., Ho, P.H., Nicollin, X., Olivero, A., Sifakis, J., Yovine, S.: The algorithmic analysis of hybrid systems. TCS 138, 3–34 (1995)
3. Alur, R., Dill, D.L.: A theory of timed automata. TCS 126, 183–235 (1994)
4. Alur, R., Henzinger, T.A., Lafferriere, G., Pappas, G.J.: Discrete abstractions of hybrid systems. Proceedings of the IEEE 88(7), 971–984 (2000)
5. Asarin, E., Maler, O., Pnueli, A.: Reachability analysis of dynamical systems having piecewise-constant derivatives. TCS 138(1), 35–65 (1995)
6. Basu, S., Pollack, R., Roy, M.F.: Algorithms in Real Algebraic Geometry. Springer (2006)
7. Ben-Or, M., Kozen, D., Reif, J.: The complexity of elementary algebra and geometry. In: Proceedings of the Sixteenth Annual ACM Symposium on Theory of Computing. pp. 457–464. STOC '84, ACM (1984)
8. Bérard, B., Haddad, S.: Interrupt timed automata. In: Proc. of FoSSaCS'09. LNCS, vol. 5504, pp. 197–211. Springer, York, UK (Mar 2009)
9. Bérard, B., Haddad, S., Jovanovič, A., Lime, D.: Parametric interrupt timed automata. In: Proceedings of the 7th Workshop on Reachability Problems in Computational Models (RP'13). LNCS, vol. 8169, pp. 59–69. Springer (2013)
10. Bérard, B., Haddad, S., Sassolas, M.: Interrupt timed automata: Verification and expressiveness. Formal Methods in System Design 40(1), 41–87 (Feb 2012)
11. Bérard, B., Haddad, S., Picaronny, C., Safey El Din, M., Sassolas, M.: Polynomial interrupt timed automata. CoRR abs/1504.04541 (Apr 2015)
12. Cassez, F., Larsen, K.G.: The impressive power of stopwatches. In: Proc. of CONCUR'00. LNCS, vol. 1877, pp. 138–152. Springer (Aug 2000)
13. Collins, G.E.: Quantifier elimination for real closed fields by cylindrical algebraic decompostion. In: Automata Theory and Formal Languages 2nd GI Conference, LNCS, vol. 33, pp. 134–183. Springer Berlin Heidelberg (1975)
14. Grossman, R., Nerode, A., Ravn, A., Rischel, H. (eds.): Hybrid systems, LNCS, vol. 736. Springer (1993)
15. Henzinger, T.A., Kopke, P.W., Puri, A., Varaiya, P.: What's decidable about hybrid automata? J. Comput. Syst. Sci. 57(1), 94–124 (1998)
16. Hong, H., Din, M.S.E.: Variant quantifier elimination. Journal of Symbolic Computation 47(7), 883–901 (2012)
17. Lafferriere, G., Pappas, G.J., Sastry, S.: O-minimal hybrid systems. MCSS 13(1), 1–21 (2000)
18. Miller, D.J.: Constructing o-minimal structures with decidable theories using generic families of functions from quasianalytic classes. ArXiv e-prints 1008.2575 (Aug 2010)
19. Miller, J.S.: Decidability and complexity results for timed automata and semi-linear hybrid automata. In: HSCC'00. LNCS, vol. 1790, pp. 296–309. Springer (2000)
20. Müller-Olm, M., Seidl, H.: Computing polynomial program invariants. Inf. Process. Lett. 91(5), 233–244 (2004)