Revisiting Reachability in Polynomial Interrupt Timed Automata

Béatrice Bérard, Serge Haddad

Revisiting Reachability in Polynomial Interrupt Timed Automata

Béatrice Bérard[a], Serge Haddad[b]

[a]Sorbonne Université, CNRS, LIP6, F-75005 Paris, France
[b]LSV, ENS Paris-Saclay, CNRS, Inria, Université Paris-Saclay, France

Abstract

Polynomial Interrupt Timed Automata (PolITA) are finite automata with clocks organized along hierarchical levels. These clocks are equipped with an interruption mechanism, well suited to the modeling of real-time operating systems. Moreover, transitions between states contain polynomial guards and updates. The reachability problem in this class is known to be in 2EXPTIME with a decision procedure based on the cylindrical algebraic decomposition. We improve this complexity to EXPSPACE mainly using a combinatorial argument and we include a reduction leading to a PSPACE lower bound.

Keywords: Interrupt timed automata, Reachability problems, Complexity

1. Introduction

Interrupt Timed Automata. Timed Automata (TA) [1] are appropriate models for the specification and verification of real-time systems. However, adding *stopwatches*, *i.e.*, clocks which can be suspended, to TA, makes all the relevant verification problems undecidable [2]. In order to address this issue, the model of Interrupt Timed Automata (ITA) was introduced in [3]. This model combines a finite set of control states, organized along hierarchical levels with exactly one *active* clock per level. In a state, only this active clock evolves with time elapsing while all other clocks are frozen, those of levels greater than the current one being null. The transitions between states are guarded by affine constraints on clocks from levels less than or equal to the current level. When a transition is increasing the current level or remaining at the same level, the active clock can be updated by an affine combination of clocks from lower levels.

Reachability for ITA is in NEXPTIME and model checking is also decidable for several classes of timed temporal logics [4, 5]. It was also proved that TA and ITA do not accept the same timed languages.

Parametric Interrupt Timed Automata. Sometimes the clock constraints of a system are only partially known. An elegant way to deal with this issue consists in introducing parameters. In TA, introduction of parameters yields undecidability results [6]. In contrast, for Parametric Interrupt Timed Automata (PITA) proposed in [7, 8] several problems including reachability remain decidable. However the complexity of solving reachability increases to 2EXPTIME when parameters occur as additive terms in guards and updates. It increases again to 2EXPSPACE when parameters may also occur as multiplicative terms.

Polynomial Interrupt Timed Automata. To tackle the problem of parameters for ITA, the authors of [9, 10] chose another direction: They consider parameters as static clocks. The guards (resp. updates) are also extended: They are defined by arbitrary polynomial constraints on clocks with levels lower than or equal to (resp. lower than) the current level. This yields the class of Polynomial Interrupt Timed Automata (PolITA). This class simultaneously extends the class of PITA and simplifies its syntax (parameters are static).

Before formally defining PolITA, we present a toy example (from [10]) describing the landing of a rocket. In the first stage, the rocket approaches the land from distance d, under gravitation g. In the second stage, the rocket is subject to a constant deceleration $h < 0$. Finally, it must reach the land with a small non negative speed (less than some fixed ε). The problem is the following: For all $g \in [7, 10]$, does there exist an $h \in [-3, -1]$ such

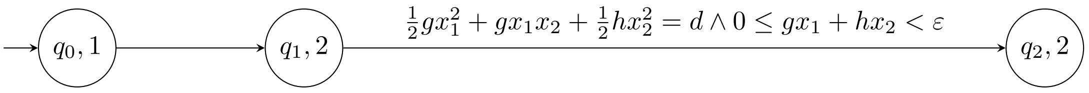

$q_0, 1$
$q_1, 2$
$\frac{1}{2}gx_1^2 + gx_1x_2 + \frac{1}{2}hx_2^2 = d \wedge 0 \leq gx_1 + hx_2 < \varepsilon$
$q_2, 2$

Figure 1: A PolITA modeling the rocket landing.

that the rocket is landing without crash? Observe that this problem is harder than reachability since it includes an alternation of quantifiers. The polynomial interrupt automaton in Figure 1 illustrates this example. The first stage is modeled by state q_0 at level 1 where the system spends x_1 time units. Then the system spends x_2 time units in q_1, representing the second stage (deceleration). The target state q_2 is reached if the rocket can land safely.

It is shown in [9, 10] that reachability and extended problems like the one above are decidable in 2EXPTIME for PolITA. The proof relies on the construction of a finite abstraction, obtained via a *cylindrical algebraic decomposition* [11, 12] related to the first order theory of reals.

Contribution. We propose an EXPSPACE procedure solving the reachability problem for PolITA. Our procedure combines a combinatorial argument about the length of runs witnessing reachability, with the resolution of the truth problem in the existential first-order theory of reals [13]. We also include a reduction showing that reachability is PSPACE-hard.

Organisation. In Section 2, we recall the syntax and semantics of PolITA, illustrated with an example. For reachability, we establish the PSPACE lower bound in Section 3 and the improved upper bound in Section 4. The corresponding procedure is illustrated in Section 5 on another example. Finally, in Section 6, we conclude and give some perspectives to this work.

2. Model

We denote respectively by $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$ and $\mathbb{R}$ the sets of natural numbers, integers, rational and real numbers. The set of non negative real numbers is denoted by $\mathbb{R}_{\geq 0}$.

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.

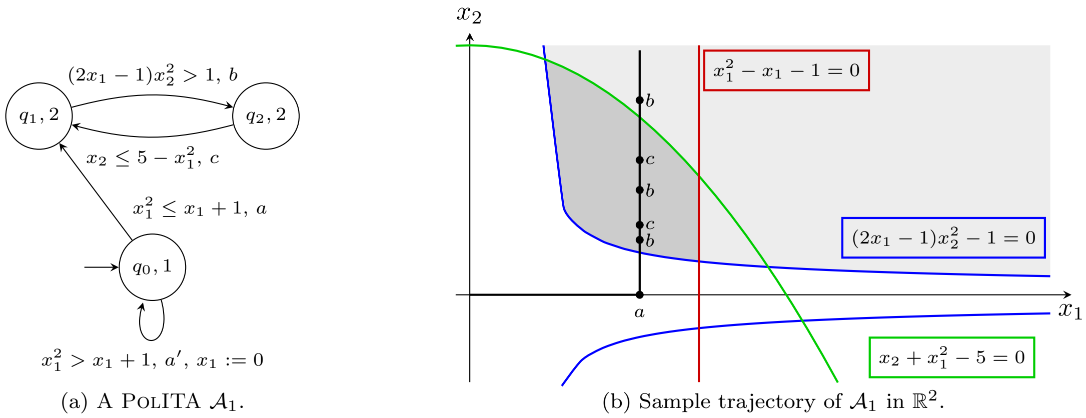
A *polynomial constraint* is a conjunction of constraints of the form $P \bowtie 0$ where $P \in \mathbb{Q}[x_1, \ldots, x_n]$ and $\bowtie \in \{<, \leq, =, \geq, >\}$, and we denote by $\mathcal{C}(X)$ the set of polynomial constraints. We also define $\mathcal{U}(X)$, the set of *polynomial updates* 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 (clock) valuation 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 combination of polynomial constraints: $v \models \varphi$ 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 $k \in \{1, \ldots, n\}$, 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 $\mathcal{A} = \langle \Sigma, Q, q_0, X, \lambda, \Delta \rangle$, 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, $q_0 \in Q$ is the initial state;
- $X = \{x_1, \ldots, x_n\}$ 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 $e = q \xrightarrow{\varphi, a, u} q'$ in Δ be a transition with q its source state, q' its target state and $tr(e) = a$ its label. The mapping λ is extended to Δ by setting $\lambda(e) = \lambda(q)$. Let $k = \lambda(q)$ 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



(a) A PoLITA $\mathcal{A}_1$.

(b) Sample trajectory of $\mathcal{A}_1$ in $\mathbb{R}^2$.

Figure 2: A PoLITA and a trajectory.

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 $i > k$, $C_i = 0$.

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 $\mathcal{T}_{\mathcal{A}} = (S, s_0, \rightarrow)$, where $S = \left\{(q, v) \mid q \in Q,\ v \in \mathbb{R}^X\right\}$ is the set of configurations, with initial configuration $s_0 = (q_0, \mathbf{0})$. The relation $\rightarrow$ 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 delay $d \in \mathbb{R}_{\geq 0}$ is defined by $(q, v) \xrightarrow{d} (q, v')$ with $v' = v +_{\lambda(q)} d$.
Discrete steps: There is a discrete step $(q, v) \xrightarrow{e} (q', v')$ for transition $e = q \xrightarrow{\varphi, a, u} q'$ in Δ, if $v \models \varphi$ and $v' = v[u]$.

A run ρ of $\mathcal{A}$ is a finite path in the graph $\mathcal{T}_{\mathcal{A}}$ alternating time and discrete steps starting from s_0. Its length $|\rho|$ is the number of discrete steps and its duration $Dur(\rho)$ is the sum of all delays. Let $\rho = (d_0, e_1, d_1, \ldots, e_n, d_n)$ be a run denoted by its sequence of steps. We denote by $last(\rho)$ the target state of transition e_n and we define $tr(\rho) \in \Sigma^*$, its label, by $tr(\rho) = tr(e_1) \ldots tr(e_n)$. We are now in position to define the language of a PoLITA.

Definition 3. Let $\mathcal{A} = \langle \Sigma, Q, q_0, X, \lambda, \Delta \rangle$ be a PoLITA and let $q_f \in Q$. The (untimed) language $\mathcal{L}(\mathcal{A}, q_f) \subseteq \Sigma^*$ is defined by:

$$\mathcal{L}(\mathcal{A}, q_f) = \{tr(\rho) \mid \rho \text{ run of } \mathcal{A} \wedge last(\rho) = q_f\}.$$

For the PoLITA $\mathcal{A}_1$ depicted in Figure 2, borrowed from [9, 10], the transition from q_0 to q_1 can only be fired before (or when) x_1 reaches $\frac{1+\sqrt{5}}{2}$, the positive root of the polynomial $A = x_1^2 - x_1 - 1$. 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 or on the green curve. The run depicted on the right is:
$\rho = (q_0, (0, 0)) \xrightarrow{1.2} (q_0, (1.2, 0)) \xrightarrow{a}$
$(q_1, (1.2, 0)) \xrightarrow{1.1} (q_1, (1.2, 1.1)) \xrightarrow{b}$
$(q_2, (1.2, 1.1)) \xrightarrow{0.3} (q_2, (1.2, 1.4)) \xrightarrow{c}$
$(q_1, (1.2, 1.4)) \xrightarrow{0.7} (q_1, (1.2, 2.1)) \xrightarrow{b}$
$(q_2, (1.2, 2.1)) \xrightarrow{0.6} (q_2, (1.2, 2.7)) \xrightarrow{c}$
$(q_1, (1.2, 2.7)) \xrightarrow{1.2} (q_1, (1.2, 3.9)) \xrightarrow{b}$
$(q_2, (1.2, 3.9))$.
For this run, we have $|\rho| = 6$, $Dur(\rho) = 5.1$, $tr(\rho) = abcbcb$ and $last(\rho) = q_2$. Clock x_1 has been frozen at the value 1.2. The polynomial constraint $B > 0$ with $B = (2x_1 - 1)x_2^2 - 1$ which guards transition b, becomes $x_2^2 > \frac{1}{1.4}$, while $C \leq 0$, with $C = x_2 - 5 - x_1^2$, which guards transition c, becomes $x_2 \leq 3.56$.

Given a PoLITA $\mathcal{A}$ and a state q the *reachability problem* asks whether there exists a valuation v and a run of $\mathcal{A}$ from $(q_0, \mathbf{0})$ to (q, v). A variant of this problem where the target is given by a state and a polynomial constraint φ on v can be handled by adding from q a transition guarded by φ to a new state q_f and checking reachability of q_f.

The reachability problem can be restated as the *emptiness problem*: Given a PolITA $\mathcal{A}$ and a state q the emptiness problem asks whether $\mathcal{L}(\mathcal{A}, q) = \emptyset$. A closely related problem is the *word problem*: Given a PolITA $\mathcal{A}$, a state q and a word $w \in \Sigma^*$, the word problem asks whether $w \in \mathcal{L}(\mathcal{A}, q)$. The word problem is in fact a particular case of the following problem: Given a PolITA $\mathcal{A}$, a state q and a finite automaton $\mathcal{B}$ over Σ, the *regular intersection problem* asks whether $\mathcal{L}(\mathcal{A}, q) \cap \mathcal{L}(\mathcal{B}) = \emptyset$. Observe that by a standard synchronized product construction, one can build in polynomial time an automaton $\mathcal{A}'$ with a target state q' such that $\mathcal{L}(\mathcal{A}, q) \cap \mathcal{L}(\mathcal{B}) = \mathcal{L}(\mathcal{A}', q')$ thus reducing the regular intersection problem to the reachability problem.

The finite abstraction of $\mathcal{T}_{\mathcal{A}}$ proposed in [9] for reachability is consistent with time elapsing, discrete jumps through the crossing of transitions, and keeps constant the truth value of constraints $P \bowtie 0$. In the resulting model, a state consists of a control state coupled with a *cell* of an appropriate *cylindrical algebraic decomposition* [11, 12]. This abstraction gives a 2EXPTIME procedure for the reachability problem.

3. A PSPACE lower bound for reachability

It is known since [14] that the truth problem for the existential theory of reals is PSPACE-complete. Figure 3 illustrates a simple reduction of the truth of a first order formula $\exists x_1 \dots \exists x_n \varphi$ to reachability of the gray state in the PolITA. The automaton has $n+1$ levels in order to choose non negative values for clocks $x_0, \dots, x_{n-1}$ and the line at level $n+1$ is used to choose the signs of $x_1, \dots, x_n$. Assuming φ is in disjunctive normal form, the arrow leading to this gray state represents in fact several arrows, one for each term of the disjunction.

Proposition 4. *The reachability problem for* PolITA *is* PSPACE*-hard.*

4. An EXPSPACE reachability procedure

The EXPSPACE procedure is inspired from the one for ITA. It is based on a combinatorial argument, with the aim of bounding the length of a run witnessing reachability. Observe that, while Lemma 5 will later be applied to runs, the sequence $(e_1, \dots, e_\ell)$ of transitions considered in this lemma is not necessarily a path in automaton $\mathcal{A}$.

Lemma 5 (Counting Lemma). *Let $\mathcal{A}$ be a* PolITA *with E transitions and n clocks. Then in a sequence $(e_1, \dots, e_\ell)$ of transitions of $\mathcal{A}$ where $\ell > (E+n)^{2n}$, there exist $i < j$ with $e_i = e_j$ such that the level of any transition e_k with $i \leq k \leq j$ is greater than or equal to the level of e_i, say p, and:*

1. *either e_i updates x_p,*
2. *or no e_k with $i \leq k \leq j$ updates x_p.*

Proof. Assume that the conclusions of the lemma are not satisfied by a sequence $\sigma = (e_1, \dots, e_\ell)$. We claim that $\ell \leq (E+n)^{2n}$. For the sake of uniformity, we enlarge σ as $\hat{\sigma}$ by adding dummy transitions e_0 at the beginning and $e_{\ell+1}$ at the end, with $\lambda(e_0) = \lambda(e_{\ell+1}) = 0$.

Step 1. We first establish a result about particular subsequences of $\hat{\sigma}$. Let $1 \leq m \leq n$ and consider a subsequence $\sigma' = (e_i, e_{i+1}, \dots, e_j)$ such that:

- $\lambda(e_i) < m$, $\lambda(e_j) < m$,
- and for all $i < k < j$, $\lambda(e_k) \geq m$.

Let $U = \{k \mid i < k < j \wedge e_k \text{ updates } x_m\}$. Since σ does not fufill Assertion 1, $|U| \leq E$. Since σ does not fufill Assertion 2, between two transitions indexed by consecutive items of U (or before the first or after the last), there can be no more than E transitions of level m that do not update x_m. Summing up, there can be no more than $E(E+1) \leq (E+1)^2$ transitions of level m that occur in σ'.

Step 2. Now we prove by induction that the number of transitions at level less than or equal to m in σ is at most $(E+m)^{2m}$. This is true for $m = 1$ by the previous proof applied to $\hat{\sigma}$. Assume the formula valid for any $m' \leq m$. Let $U_m = \{k \mid \lambda(e_k) \leq m\}$ and consider any subsequence σ' occurring between two transitions indexed by consecutive items of U_m (or before the first or after the last). Then, the number of transitions of level $m+1$ in σ' is less than or equal than $(E+1)^2$.

Then grouping the transitions of level $m+1$ between the occurrences of transition of lower level we obtain that the number of transitions at levels less than or equal to $m+1$ is at most:

$$\begin{aligned}(E+m)^{2m} + ((E+m)^{2m}+1)(E+1)^2 \leq \\ (E+m)^{2m+2} + 2(E+m)^{2m} \leq \\ (E+m+1)^{2(m+1)}\end{aligned}$$

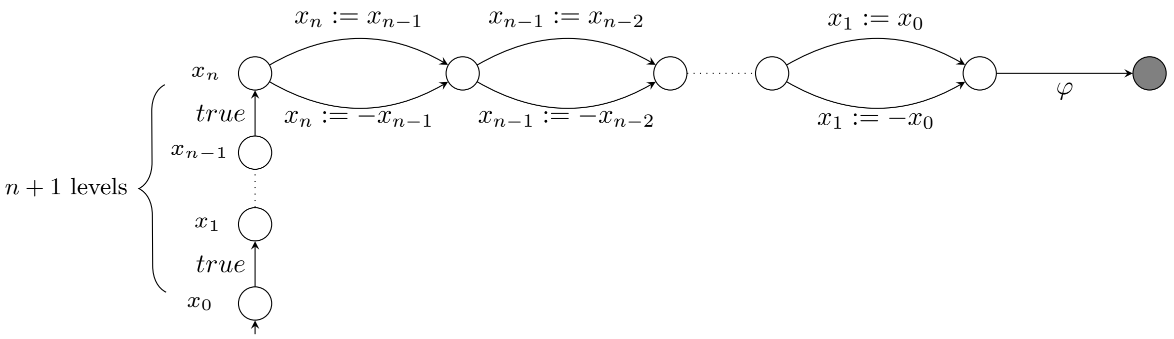


Figure 3: A PolITA for the reduction of the existential theory of reals.

□

Building on the lemma above, we show that reachability can be checked over runs with length less than or equal to $(E+n)^{2n}$, where E is the number of transitions and n is the number of clocks, which yields:

Proposition 6. *The reachability problem for* PolITA *belongs to* EXPSPACE*.*

Proof. Let $\mathcal{A} = (\Sigma, Q, q_0, X, \lambda, \Delta)$ be a PolITA with n clocks, let $q_f \in Q$ be a target state and let $E = |\Delta|$ be the number of transitions of $\mathcal{A}$. Consider a run of minimal length ρ from $(q_0, \mathbf{0})$ to a configuration (q_f, v_f) for some v_f. Writing $(e_1, \ldots, e_\ell)$ for the sequence of transitions underlying ρ, we suppose now that $\ell > B = (E+n)^{2n}$. We build a run ρ' from $(q_0, \mathbf{0})$ to (q_f, v_f) that is strictly smaller, hence contradicting the minimality hypothesis.

Since $\ell > B$, one of the two cases of Lemma 5 applies. Therefore there are two indices $i < j$ such that $e_i = e_j$ that we denote by e in the sequel, at level k. Thus ρ can be written as $\rho_1 e \hat{\rho} e \rho_2$ where the subrun $\hat{\rho}$ contains only transitions of level higher than or equal to k. Moreover:

1. Either e updates x_k. In this case, all clocks have the same value after the first and the second occurrence of e. Hence removing $e\hat{\rho}$ from ρ yields a valid run ρ' of $\mathcal{A}$ reaching (q_f, v_f). Run ρ' is strictly smaller than ρ.
2. Either no update occurred for x_k in $e\hat{\rho}$. In this case, upon reaching the second occurrence of e, the clocks of level $i < k$ have retained the same value, while x_k has increased by $d = Dur(\hat{\rho}) = Dur(e\hat{\rho})$. Hence when replacing $e\hat{\rho}$ by a time step of duration d, the resulting configuration is unchanged. This also yields a shorter run.

The decision procedure works as follows. It non deterministically guesses a path in the PolITA with length ℓ less than or equal to B. In order to check that this path yields a run, it builds a polynomial system with variables

$$\{x_i^j \mid 1 \leq i \leq n, 1 \leq j \leq \ell\} \cup \{d_j, 1 \leq j \leq \ell\}$$

where x_i^j is the value of clock x_i after the jth step, and d_j is the delay before the jth discrete transition. The equations and inequations are deduced from the guards and updates of discrete transitions in the path and the delays. The size of this system is exponential w.r.t. the size of the PolITA.

As such a system can be solved in polynomial space according to Canny [13], we obtain a procedure in NEXPSPACE= EXPSPACE. □

Using the observation of section 2, one immediately gets:

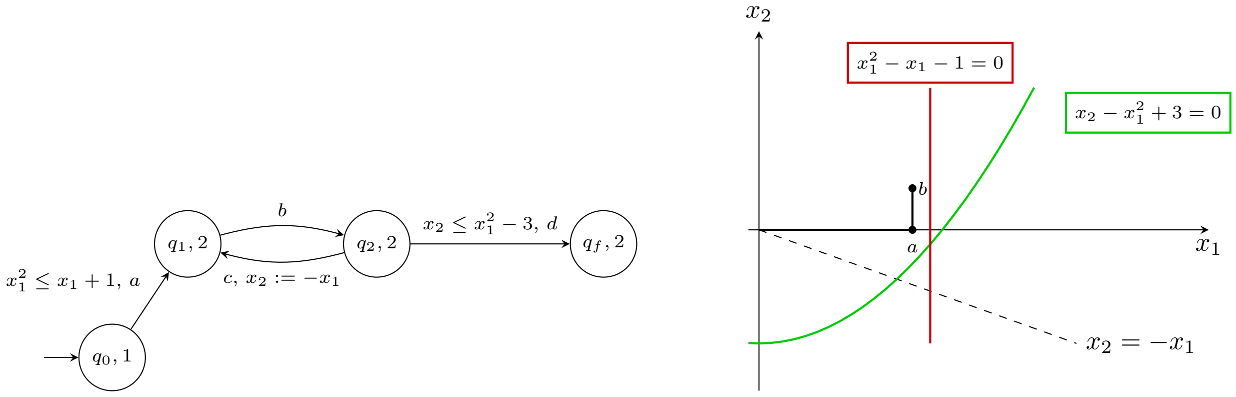
Corollary 7. *The regular intersection problem for* PolITA *belongs to* EXPSPACE*.*

Observe that, given a PolITA without silent transitions (*i.e.*, labelled by ε), and a word w, we can use the technique above, guess a path with length $|w|$ and solve the associated polynomial system. This gives a NPSPACE=PSPACE upper bound for the word problem. Furthermore, using Proprosition 4 which is also valid for the word problem, we obtain:

Proposition 8. *The word problem for* PolITA *without silent transitions is* PSPACE*-complete.*

5. Illustration

Let us illustrate the polynomial equation systems associated with two guesses on the example depicted in Figure 4a with q_f as target state.



(a) A PoLITA $\mathcal{A}_2$.

(b) Geometric view of path ab.

Figure 4: A PoLITA with geometric view.

The shortest path from q_0 to q_f corresponds to the word *abd*. We obtain the following polynomial system where equations (1,2,3) (resp. (4,5,6), (7,8,9)) correspond to the transition labeled by a (resp. b, d).

$$x_1^1 = d_1 \tag{1}$$
$$x_2^1 = 0 \tag{2}$$
$$d_1^2 \le d_1 + 1 \text{ and } d_1 \ge 0 \tag{3}$$
$$x_1^2 = d_1 \tag{4}$$
$$x_2^2 = d_2 \tag{5}$$
$$d_2 \ge 0 \tag{6}$$
$$x_1^3 = d_1 \tag{7}$$
$$x_2^3 = d_2 + d_3 \tag{8}$$
$$d_2 + d_3 \le d_1^2 - 3 \text{ and } d_3 \ge 0 \tag{9}$$

This system has a partial solution over variables x_1^1, x_1^2, d_1, x_1^2, x_2^2, d_2 with transitions a and b illustrated in Figure 4b. However, there is no global solution because equations (3), (6) and (9) imply:

- $d_1 \le \frac{1+\sqrt{5}}{2} < 2$,
- and $0 \le d_2 + d_3 \le d_1^2 - 3 \le d_1 - 2$.

The second shortest path corresponds to the word *abcbd*, using one iteration of the loop. Equations (1,2,3) (resp. (4,5,6), (7,8,9), (10,11), (12,13,14)) correspond to the transition labeled by a (resp. b, c, b, d).

$$x_1^1 = d_1 \tag{1}$$
$$x_2^1 = 0 \tag{2}$$
$$d_1^2 \le d_1 + 1 \text{ and } d_1 \ge 0 \tag{3}$$
$$x_1^2 = d_1 \tag{4}$$
$$x_2^2 = d_2 \tag{5}$$
$$d_2 \ge 0 \tag{6}$$
$$x_1^3 = d_1 \tag{7}$$
$$x_2^3 = d_2 + d_3 \tag{8}$$
$$d_3 \ge 0 \tag{9}$$
$$x_1^4 = d_1 \tag{10}$$
$$x_2^4 = -d_1 \tag{11}$$
$$x_1^5 = d_1 \tag{12}$$
$$x_2^5 = -d_1 + d_4 \tag{13}$$
$$-d_1 + d_4 \le d_1^2 - 3 \text{ and } d_4 \ge 0 \tag{14}$$

This second system has a solution for any value of $d_4 \le d_1^2 + d_1 - 3$ provided $\frac{-1+\sqrt{13}}{2} \le d_1 \le \frac{1+\sqrt{5}}{2}$. The value $\frac{-1+\sqrt{13}}{2}$ is the positive x_1-coordinate for the intersection of the green curve with the dashed diagonal. A trajectory, for which the final b and d steps must stay below the green curve is depicted with a zoom on Figure 5.

6. Conclusion

As described in the introduction, PoLITA presented several advantages over PITA: extension of expressiveness, simplified syntax and same complexity for reachability based problems

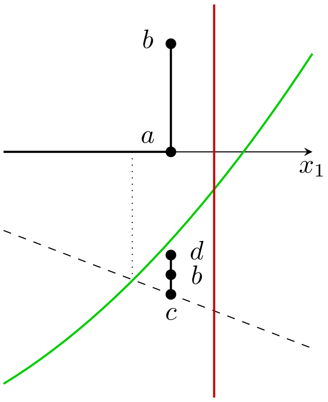


Figure 5: Zoom for path *abcbd*.

(2EXPTIME). Our new decision procedure for reachability in PolITA with reduced complexity (EXPSPACE) exhibits an additional interest.

There is still a big gap between this upper bound and the PSPACE lower bound but thanks to the reduction from the existential theory of reals, any improvement of the lower bound for this problem will transfer to reachability in PolITA.

We are now looking for the specification of a restricted temporal logic for which we could design a decision procedure for the model checking problem with the same complexity.

Acknowledgement. We thank the reviewer for a careful reading and in particular for the suggestion about the lower bound which led to a complete complexity characterization in Proposition 8.

[1] R. Alur, D. L. Dill, A theory of timed automata, Theoretical Computer Science 126 (1994) 183–235.

[2] T. A. Henzinger, P. W. Kopke, A. Puri, P. Varaiya, What's decidable about hybrid automata?, Journal of Computer and System Science 57 (1) (1998) 94–124.

[3] B. Bérard, S. Haddad, Interrupt timed automata, in: Proceedings of FOSSACS 2009, Vol. 5504 of LNCS, Springer, 2009, pp. 197–211.

[4] B. Bérard, S. Haddad, M. Sassolas, Real time properties for interrupt timed automata, in: Proceedings of TIME 2010, IEEE Computer Society, 2010, pp. 69–76.

[5] B. Bérard, S. Haddad, M. Sassolas, Interrupt timed automata: verification and expressiveness, Formal Methods Syst. Des. 40 (1) (2012) 41–87.

[6] J. S. Miller, Decidability and complexity results for timed automata and semi-linear hybrid automata, in: Proceedings of HSCC'00, Vol. 1790 of LNCS, Springer, 2000, pp. 296–309.

[7] B. Bérard, S. Haddad, A. Jovanovic, D. Lime, Parametric interrupt timed automata, in: Proceedings of RP 2013, Vol. 8169 of LNCS, Springer, 2013, pp. 59–69.

[8] B. Bérard, S. Haddad, A. Jovanovic, D. Lime, Interrupt timed automata with auxiliary clocks and parameters, Fundam. Inform. 143 (3-4) (2016) 235–259.

[9] B. Bérard, S. Haddad, C. Picaronny, M. Safey El Din, M. Sassolas, Polynomial Interrupt Timed Automata, in: Proceedings of RP'15, Vol. 9328 of LNCS, Springer, 2015, pp. 20–32.

[10] B. Bérard, S. Haddad, C. Picaronny, M. Safey El Din, M. Sassolas, Polynomial interrupt timed automata: Verification and expressiveness, Information and Computation 277 (2021) 104580. doi:10.1016/j.ic.2020.104580.

[11] G. E. Collins, Quantifier elimination for real closed fields by cylindrical algebraic decomposition, in: Automata Theory and Formal Languages 2nd GI Conference, Vol. 33 of LNCS, Springer Berlin Heidelberg, 1975, pp. 134–183.

[12] S. Basu, R. Pollack, M.-F. Roy, Algorithms in Real Algebraic Geometry, Springer, 2006.

[13] J. Canny, Some algebraic and geometric computations in PSPACE, in: Proceedings of the Annual ACM Symposium on Theory of Computing, 1988, pp. 460–467.

[14] J. H. Reif, Complexity of the mover's problem and generalizations (extended abstract), in: 20th Annual Symposium on Foundations of Computer Science, San Juan, Puerto Rico, 29-31 October 1979, IEEE Computer Society, 1979, pp. 421–427. doi:10.1109/SFCS.1979.10.