Liveness in L/U-Parametric Timed Automata

Étienne André
Université Paris 13, LIPN
CNRS, UMR 7030, F-93430, Villetaneuse, France

Didier Lime
École Centrale de Nantes, LS2N
CNRS, UMR 6004, Nantes, France

***Abstract*—We study timed systems in which some timing features are unknown parameters. Parametric timed automata are a classical formalism for such systems but for which most interesting problems are undecidable. Lower-bound/upper-bound parametric timed automata (L/U-PTAs) achieve decidability for reachability properties by enforcing a separation of parameters used as upper bounds in the automaton constraints, and those used as lower bounds.**

We further study L/U-PTAs by considering liveness related problems. We prove that: (1) the existence of at least one parameter valuation for which there exists an infinite run in the automaton is PSPACE-complete; (2) the existence of a parameter valuation such that the system has a deadlock is however undecidable; (3) the problem of the existence of a valuation for which a run remains in a given set of locations exhibits a very thin border between decidability and undecidability.

***Index Terms*—L/U-PTA, EG-emptiness, deadlock-freeness, infinite run**

1. Introduction

Following Lamport, properties of systems are often characterized as safety properties (“something bad will never happen”) and liveness properties (“something good will eventually happen”) [Lam77]. Safety generally reduces to reachability, while liveness is more complex. The “good” behavior may not be reached for two main reasons: either there is a deadlock, a state in which the system cannot evolve anymore, or there is a livelock, an infinite path never reaching the “good” behavior. Both situations are captured by the CTL operator EG [CES86].

We study here those behaviors in the context of parametric timed systems, in which some timing features (*e. g.*, the duration of a task, a transmission delay in a network, the delay to trigger a watchdog, etc.) are not known and replaced by symbolic constants, called *parameters*. The objective of verification on such partially defined systems, is then to synthesize the possible valuations of parameters such that some properties are satisfied.

Related Works. Parametric timed automata (PTAs) [AHV93] have been introduced to deal with such parametric timed systems. They consist in finite automata equipped with real-valued clocks that can be compared with constants or parameters in constraints restricting if and when the edges can be taken.

The simple problem of whether there exists a valuation for each parameter such that some control location is reachable in the timed automaton obtained by replacing the parameters with those valuations (also called EF-emptiness) is undecidable for PTAs for both integer- and rational-valued parameters. Several alternative proofs refine this result in terms of the number of parameters, number of clocks compared to parameters, types of constraints, etc. (see, *e. g.*, [Mil00], [Doy07], [BO14], [BBLS15], [And15]).

In order to overcome these disappointing results, lower-bound/upper-bound parametric timed automata (L/U-PTAs) are introduced as a subclass of PTAs where each parameter either always appears as an upper bound when compared to a clock, or always as a lower bound [HRSV02]. The EF-emptiness problem, and also the EF-universality problem (“Can we reach a given location, regardless of what valuations we give to the parameters?”) are decidable for L/U-PTAs.

In [BL09], infinite acceptance properties are considered: the emptiness and the universality of the valuation set for which a given location is infinitely often traversed are decidable for integer-valued parameters.

In [JLR15], it is shown that the AF-emptiness problem (“Is the set of parameter valuations such that the system reaches a given location for all runs, empty?”) is undecidable for L/U-PTAs with integer- and rational-valued parameters.

Contribution. With the notable exception of [JLR15], and to some extent of [BL09] which addresses the existence of cycles, all the works cited above focus on safety properties, through the basic problem of reachability. This is maybe not so surprising given that most results related to this simpler problem are already negative.

We nonetheless address here the problem of liveness in PTAs, and more precisely, with the negative result

This work is partially supported by the ANR national research program “PACS” (ANR-14-CE28-0002). This manuscript is a slightly extended and modified version of the paper of the same name published in the proceedings of the 17th International Conference on Application of Concurrency to System Design (ACSD 2017). The final conference version is available on IEEE explore.

of [JLR15] on AF-emptiness in mind, we start from L/U-PTAs with rational-valued parameters and further refine both the model and the properties. We prove that:

1) deciding the existence of at least one parameter valuation for which there exists an infinite run (discrete cycle) in the automaton is PSPACE-complete;
2) deciding the existence of a parameter valuation such that the system has a deadlock is however undecidable;
3) the problem of the existence of a valuation for which a run remains in a given set of locations exhibits a very thin border between decidability and undecidability: while this problem is decidable for L/U-PTAs with a bounded parameter domain with closed bounds, it becomes undecidable if either the assumption of boundedness or of closed bounds is lifted. This result confirms that L/U-PTAs stand at the border between decidability and undecidability.

Differently from [BL09], we use here no accepting locations. In addition, our parameters are not restricted to be integer-valued, and can be rational-valued.

Outline. We recall the necessary preliminaries in Section 2. We then consider the problem of the existence of at least one parameter valuation for which there exists an infinite run (Section 3), for which there exists a deadlock (Section 4), and for which a run remains in a given set of locations (Section 5). We conclude and discuss perspectives in Section 6.

2. Preliminaries

2.1. Clocks, Parameters and Constraints

Let $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}_+$ and $\mathbb{R}_+$ denote the sets of non-negative integers, integers, non-negative rational numbers and non-negative real numbers respectively. Let $\mathcal{I}(\mathbb{N})$ denote the set of non-necessarily closed intervals on $\mathbb{N}$, *i. e.,* the set of intervals of the form $[a, b]$, $(a, b]$, $[a, b)$ or (a, b) where $a, b \in \mathbb{N}$ and $a \leq b$.

Throughout this paper, we assume a set $X = \{x_1, \dots, x_H\}$ of *clocks*, *i. e.,* real-valued variables that evolve at the same rate. A clock valuation is a function $w : X \to \mathbb{R}_+$. We identify a clock valuation w with the point $(w(x_1), \dots, w(x_H))$ of $\mathbb{R}_+^H$. We write $\vec{0}$ for the clock valuation that assigns 0 to all clocks. Given $d \in \mathbb{R}_+$, $w + d$ denotes the valuation such that $(w + d)(x) = w(x) + d$, for all $x \in X$. Given $R \subseteq X$, we define the *reset* of a valuation w, denoted by $[w]_R$, as follows: $[w]_R(x) = 0$ if $x \in R$, and $[w]_R(x) = w(x)$ otherwise.

We assume a set $P = \{p_1, \dots, p_M\}$ of *parameters*, *i. e.,* unknown constants. A parameter *valuation* v is a function $v : P \to \mathbb{Q}_+$. We identify a valuation v with the point $(v(p_1), \dots, v(p_M))$ of $\mathbb{Q}_+^M$. An *integer* parameter valuation is a valuation v such that $\forall p \in P, v(p) \in \mathbb{N}$.

In the following, we assume $\bowtie \in \{<, \leq, \geq, >\}$. Throughout this paper, lt denotes a linear term over $X \cup P$ of the form $\sum_{1 \leq i \leq H} \alpha_i x_i + \sum_{1 \leq j \leq M} \beta_j p_j + d$, with $x_i \in X$, $p_j \in P$, and $\alpha_i, \beta_j, d \in \mathbb{Z}$. A *constraint* C (*i. e.,* a convex polyhedron) over $X \cup P$ is a conjunction of inequalities of the form $lt \bowtie 0$. Given a parameter valuation v, $v(C)$ denotes the constraint over X obtained by replacing each parameter p in C with $v(p)$. Likewise, given a clock valuation w, $w(v(C))$ denotes the expression obtained by replacing each clock x in $v(C)$ with $w(x)$. We say that v *satisfies* C, denoted by $v \models C$, if the set of clock valuations satisfying $v(C)$ is nonempty. Given a parameter valuation v and a clock valuation w, we denote by $w|v$ the valuation over $X \cup P$ such that for all clocks x, $w|v(x) = w(x)$ and for all parameters p, $w|v(p) = v(p)$. We use the notation $w|v \models C$ to indicate that $w(v(C))$ evaluates to true. We say that C is *satisfiable* if $\exists w, v$ s.t. $w|v \models C$.

A *guard* g is a constraint over $X \cup P$ defined by inequalities of the form $x \bowtie \sum_{1 \leq j \leq M} \beta_j p_j + d$, with $\beta_j \in \{0, 1\}$ and $d \in \mathbb{Z}$.

2.2. Parametric Timed Automata

2.2.1. Syntax.

Definition 1. *A PTA $\mathcal{A}$ is a tuple $\mathcal{A} = (\Sigma, L, l_0, X, P, I, E)$, where: i) Σ is a finite set of actions, ii) L is a finite set of locations, iii) $l_0 \in L$ is the initial location, iv) X is a finite set of clocks, v) P is a finite set of parameters, vi) I is the invariant, assigning to every $l \in L$ a guard $I(l)$, vii) E is a finite set of edges $e = (l, g, \sigma, R, l')$ where $l, l' \in L$ are the source and target locations, $\sigma \in \Sigma$, $R \subseteq X$ is a set of clocks to be reset, and g is a guard.*

Given a parameter valuation v, we denote by $v(\mathcal{A})$ the non-parametric timed automaton where all occurrences of a parameter p_i have been replaced by $v(p_i)$.

2.2.2. Concrete Semantics.

Definition 2 (Concrete semantics of a TA)**.** *Given a PTA $\mathcal{A} = (\Sigma, L, l_0, X, P, I, E)$, and a parameter valuation v, the concrete semantics of $v(\mathcal{A})$ is given by the timed transition system $(S, s_0, \to)$, with*

- $S = \{(l, w) \in L \times \mathbb{R}_+^H \mid w|v \models I(l)\}$, $s_0 = (l_0, \vec{0})$
- *$\to$ consists of the discrete and (continuous) delay transition relations:*
  - *discrete transitions: $(l, w) \overset{e}{\to} (l', w')$, if $(l, w), (l', w') \in S$, there exists $e = (l, g, \sigma, R, l') \in E$, $w' = [w]_R$, and $w|v \models g$.*
  - *delay transitions: $(l, w) \overset{d}{\to} (l, w + d)$, with $d \in \mathbb{R}_+$, if $\forall d' \in [0, d], (l, w + d') \in S$.*

Moreover we write $(l, w) \overset{e}{\mapsto} (l', w')$ for a combination of a delay and discrete transition where $((l, w), e, (l', w')) \in \mapsto$ if $\exists d, w'' : (l, w) \overset{d}{\to} (l, w'') \overset{e}{\to} (l', w')$.

Given a TA $v(\mathcal{A})$ with concrete semantics $(S, s_0, \to)$, we refer to the states of S as the *concrete states* of $v(\mathcal{A})$. A (concrete) *run* of $v(\mathcal{A})$ is a possibly infinite alternating sequence of concrete states of $v(\mathcal{A})$ and edges starting from the initial concrete state s_0 of the form $s_0 \overset{e_0}{\mapsto} s_1 \overset{e_1}{\mapsto} \cdots \overset{e_{m-1}}{\mapsto} s_m \overset{e_m}{\mapsto} \cdots$, such that for all $i = 0, 1, \dots$, $e_i \in E$, and

$(s_i, e_i, s_{i+1}) \in \mapsto$. Given a state $s = (l, w)$, we say that s is reachable (or that $v(\mathcal{A})$ reaches s) if s belongs to a run of $v(\mathcal{A})$. By extension, we say that l is reachable in $v(\mathcal{A})$, if there exists a state (l, w) that is reachable. Given a set of locations $T \subseteq L$ (T stands for "target"), we say that a run stays in T if all of its states (l, w) are such that $l \in T$. A maximal run is a run that is either infinite (*i. e.,* contains an infinite number of discrete transitions), or that cannot be extended by a discrete transition. A maximal run is deadlocked if it is finite, *i. e.,* contains a finite number of discrete transitions. By extension, we say that a TA is deadlocked if it contains at least one deadlocked run.

2.3. Subclasses of PTAs

Let us recall L/U-PTAs [HRSV02], [BL09].

Definition 3 (L/U-PTA)**.** *An L/U-PTA is a PTA where the set of parameters is partitioned into lower-bound parameters and upper-bound parameters, where an upper-bound (resp. lower-bound) parameter p_i is such that, for every guard or invariant constraint $x \bowtie \sum_{1 \leq j \leq M} \beta_j p_j + d$, we have: $\beta_i = 1$ implies $\bowtie \in \{\leq, <\}$ (resp. $\bowtie \in \{\geq, >\}$).*

Recall from our definition of guard that β_i can only be 0 or 1.

L/U-PTAs enjoy a well-known monotonicity property recalled in the following lemma (that corresponds to a reformulation of [HRSV02, Prop 4.2]), stating that increasing upper-bound parameters or decreasing lower-bound parameters can only add behaviors.

Lemma 1. *Let $\mathcal{A}$ be an L/U-PTA and v be a parameter valuation. Let v' be a valuation such that for each upper-bound parameter p^+, $v'(p^+) \geq v(p^+)$ and for each lower-bound parameter p^-, $v'(p^-) \leq v(p^-)$. Then any run of $v(\mathcal{A})$ is a run of $v'(\mathcal{A})$.*

In this paper, we will also consider *bounded* PTAs, *i. e.,* PTAs with a bounded parameter domain that assigns to each parameter an infimum and a supremum, both integers.

Definition 4 (bounded PTA)**.** *A* bounded PTA *is $\mathcal{A}_{|bounds}$, where $\mathcal{A}$ is a PTA, and $bounds : P \rightarrow \mathcal{I}(\mathbb{N})$ assigns to each parameter p an interval* $[\inf, \sup]$, $(\inf, \sup]$, $[\inf, \sup)$, *or* $(\inf, \sup)$, *with* $\inf, \sup \in \mathbb{N}$. *We use* $\inf(p, bounds)$ *and* $\sup(p, bounds)$ *to denote the infimum and the supremum of p, respectively. (Note that we rule out ∞ as a supremum.)*

We say that a bounded PTA is a closed bounded PTA *if, for each parameter p, its ranging interval $bounds(p)$ is of the form* $[\inf, \sup]$*; otherwise it is an* open bounded PTA*.*

We define similarly bounded L/U-PTAs.

Whereas bounded PTAs are naturally a subclass of PTAs, we showed in [ALR16b] that bounded L/U-PTAs are *incomparable* with L/U-PTAs: a consequence is that undecidability results for bounded L/U-PTAs cannot be automatically extended to L/U-PTAs; conversely, decidability results for L/U-PTAs cannot be automatically extended to bounded L/U-PTAs.

2.4. Decision Problems

Let $\mathcal{P}$ be a given a class of decision problems.

$\mathcal{P}$-emptiness problem:
INPUT: A PTA $\mathcal{A}$ and an instance ϕ of $\mathcal{P}$
PROBLEM: Is the set of parameter valuations v such that $v(\mathcal{A})$ satisfies ϕ empty?

In this paper, we mainly focus on the following three decision problems:

- deadlock-existence: given a TA $v(\mathcal{A})$, is there at least one run of $v(\mathcal{A})$ that is deadlocked, *i. e.,* has no discrete successor (possibly after some delay)?
- cycle-existence: given a TA $v(\mathcal{A})$, is there at least one run of $v(\mathcal{A})$ with an infinite number of discrete transitions?
- EG[1]: given a TA $v(\mathcal{A})$ and a subset T of its locations, is there at least one maximal run of $v(\mathcal{A})$ along which the location always remain in T?

For example, given a PTA $\mathcal{A}$, deadlock-existence-emptiness asks: "is the set of parameters valuations v such that at least one run of $v(\mathcal{A})$ is deadlocked, *i. e.,* has no discrete successor (possibly after some delay), empty?". In the following, we often abbreviate deadlock-existence-emptiness and cycle-existence-emptiness as ED-emptiness and EC-emptiness, respectively.

Note that ED-emptiness is equivalent to AC-universality, where AC-universality asks whether all parameter valuations are such that all maximal runs contain an infinite number of discrete transitions. Conversely, EC-emptiness is equivalent to AD-universality (for all valuations, all runs are deadlocked). In addition, EG-emptiness is also close to both former problems: EG is true if there exists either a finite run with a deadlock staying in T, or an infinite run staying in T.

3. Cycle-Existence-Emptiness

Theorem 1. *The cycle-existence-emptiness problem is decidable for closed bounded L/U-PTAs.*

Proof. Recall that, thanks to the monotonicity property of L/U-PTAs (recalled in Lemma 1), any run possible for a valuation v of the parameters is also possible for any valuation of the parameters for which the upper-bound (resp. lower-bound) parameters are larger (resp. smaller) than or equal to that of v.

Let $\mathcal{A}_{|bounds}$ be a closed bounded L/U-PTA. Let $v_{\inf/\sup}$ be the valuation such that, for each lower-bound parameter p^-, $v_{\inf/\sup}(p^-) = \inf(p^-, bounds)$ and, for each upper-bound parameter p^+, $v_{\inf/\sup}(p^+) = \sup(p^+, bounds)$.

1) If $v_{\inf/\sup}(\mathcal{A})$ contains an infinite run (which can be checked in PSPACE [AD94], and can be performed efficiently in practice using, *e. g.,* the zone graph), then since $\mathcal{A}_{|bounds}$ is closed, $v_{\inf/\sup}$ belongs to $bounds$,

[1] The name "EG" comes from the CTL syntax, and is consistent with EF and AF used in [JLR15].

and hence the set of parameter valuations that yield an infinite run is not empty.

2) On the contrary, if $v_{\inf/\sup}(\mathcal{A})$ contains no infinite run, then from the monotonicity property of L/U-PTAs (Lemma 1), no other valuation in *bounds* gives a TA with an infinite run, as such a TA could only contain less runs. Hence the set of parameter valuations that yield an infinite run is empty.

□

The above result cannot be used as such for non-bounded L/U-PTAs as a cycle that exists for an infinite parameter valuation may not exist for any finite parameter valuation: consider the L/U-PTA in Figure 1b. This L/U-PTA has an infinite run for $p = \infty$, but for any parameter valuation (*i. e.,* different from ∞), the number of self-loops in l_0 is bounded by p, and hence finite. However, extending to rational-valued parameters a result from [BL09], we can still prove decidability.

Lemma 2. *Given an L/U-PTA $\mathcal{A}$ and a subset of its locations T, the problem of the existence of at least one parameter valuation v such that $v(\mathcal{A})$ has a run passing infinitely often through T is PSPACE-complete.*

Proof. Let us prove that there exists a rational-valued valuation satisfying the property iff there exists an integer-valued valuation doing so.

⇐ Considering an integer valuation is also a rational-valued valuation, the result trivially holds.

⇒ Assume there exists a rational-valued parameter valuation v for which $v(\mathcal{A})$ contains an infinite run passing infinitely often through locations of T. Let v' be the integer parameter valuation obtained from v as follows:

$$v'(p) = \begin{cases} v(p) & \text{if } v(p) \in \mathbb{N} \\ \lceil v(p) \rceil & \text{if } p \text{ is an upper-bound parameter} \\ \lfloor v(p) \rfloor & \text{if } p \text{ is a lower-bound parameter} \end{cases}$$

From the monotonicity property of L/U-PTAs (Lemma 1), if $v(\mathcal{A})$ yields an infinite run passing infinitely often through locations of T, then $v'(\mathcal{A})$ does too.

Observe that this is not true for general PTAs: in Figure 1a, there is an infinite run passing infinitely often through l_0 iff $0 < p < 1$; therefore, there exist rational-valued valuations satisfying the property, but no integer-valued valuation.

Now, in [BL09, Theorem 8], it is proved that the problem of the emptiness of the set of integer parameter valuations for which there exists an infinite run passing infinitely often through T is PSPACE-complete. This concludes the proof. □

Theorem 2. *The cycle-existence-emptiness problem is decidable and PSPACE-complete for L/U-PTAs.*

Proof. Let $\mathcal{A}$ be an L/U-PTA. The set of parameter valuations for which $\mathcal{A}$ has an infinite run is empty iff the set of parameter valuations for which $\mathcal{A}$ has an infinite run passing infinitely often through L (where L denotes all locations of $\mathcal{A}$) is empty. Hence we can directly apply our intermediate Lemma 2 to conclude that this problem is decidable and PSPACE-complete. □

Without surprise (with the rule of thumb that any non-trivial problem for PTAs is undecidable), this problem becomes undecidable for general PTAs, even when bounded. We do include the full proof of this result as it will be used later on to prove more subtle results.

Theorem 3. *The cycle-existence-emptiness problem is undecidable for (bounded) PTAs with 3 clocks and 1 parameter.*

Proof. We reduce from the boundedness problem of a 2-counter machine, which is undecidable [Min67].

Recall that a deterministic 2-counter machine has two non-negative counters C_1 and C_2, a finite number of states and a finite number of transitions, which can be of the form:

- when in state q_i, increment C_k and go to q_j;
- when in state q_i, if $C_k = 0$ then go to q_k, otherwise go to q_j.

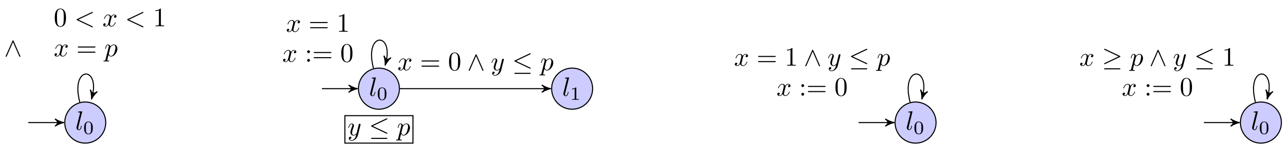
The machine starts in state q_0 with the counters set to 0; by definition, it halts when it reaches a specific state called $\mathsf{q}_{\mathsf{halt}}$. The boundedness problem for 2-counter machines asks whether, along the unique maximal run, the value of the counters remains smaller than some bound, and is undecidable [Min67].

Given such a machine $\mathcal{M}$, we encode it as a PTA $\mathcal{A}(\mathcal{M})$; our encoding is inspired by an existing encoding of a 2-counter machine, used to (re)prove the undecidability of the EF-emptiness problem for bounded PTAs and then further related results, and found in [ALR16a]. However, we had to modify it in two directions:*i*) we adapt the construction so that it fits the cycle-existence-problem instead of the EF-emptiness problem; and *ii*) we change the model of the 2-counter machine (in [ALR16a], we used the model of [AHV93], where the machine has three instructions: increment, decrement, zero-test (and block if unsatisfied)). This second part required us to modify the gadgets.

Let us now describe this encoding in details, as we will modify it in the subsequent proofs.

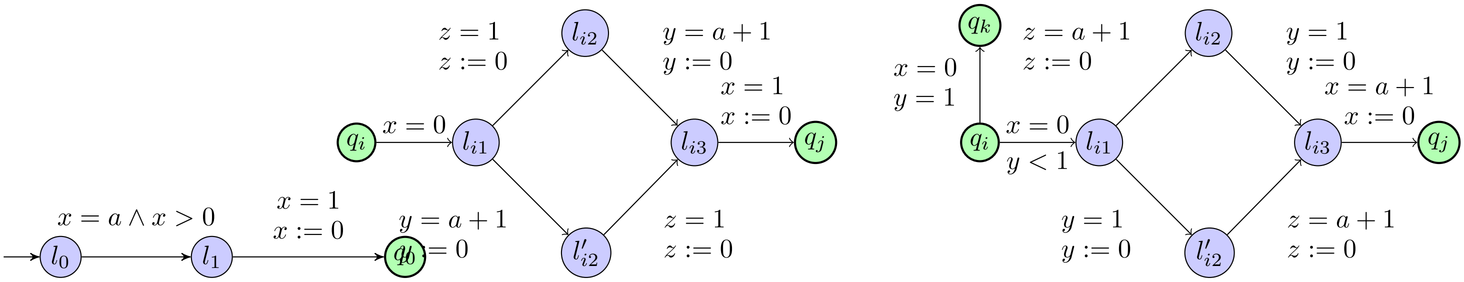
Each state q_i of the machine is encoded as a location of the automaton, which we call q_i. The counters are encoded using clocks x, y and z and one parameter a, with the following relations with the values c_1 and c_2 of counters C_1 and C_2: when $x = 0$, we have $y = 1 - ac_1$ and $z = 1 - ac_2$. All three clocks are parametric, *i. e.,* are compared with a in some guard or invariant of the encoding. We will see that a is a rational-valued bounded parameter, typically in $[0, 1]$ (although not bounding a has no impact on the proof).

We initialize the clocks with the gadget in Figure 2a (that also blocks the case where $a = 0$). Note that, throughout the paper, we highlight in thick green the locations of the PTA corresponding to a state of the 2CM (in contrast with other locations added in the encoding to maintain the matching between the clock values and the counter values). Since all clocks are initially 0, in Figure 2a clearly, when in q_0 with $x = 0$, we have $y = z = 1$, which indeed corresponds to counter values 0.



(a) (b) (c) (d)

Figure 1: Examples of PTA (a) and L/U-PTAs (b–d)



(a) Initial gadget (b) Increment gadget (C_1) (c) 0-test and decrement gadget (C_1)

Figure 2: EC-emptiness: gadgets

We now present the gadget encoding the increment instruction of C_1 in Figure 2b. The transition from q_i to l_{i1} only serves to clearly indicate the entry in the increment gadget and is done in 0 time unit. Since we use only equalities, there are really only two paths that go through the gadget: one going through l_{i2} and one through l'_{i2}. Let us begin with the former. We start from some encoding configuration: $x = 0$, $y = 1 - ac_1$ and $z = 1 - ac_2$ in q_i (and therefore the same in l_{i1}). We can enter l_{i2} (after elapsing enough time) if $1 - ac_2 \leq 1$, *i. e.,* $ac_2 \geq 0$, which implies that $a \geq 0$, and when entering l_{i2} we have $x = ac_2$, $y = 1 - ac_1 + ac_2$ and $z = 0$. Then we can enter l_{i3} if $1 - ac_1 + ac_2 \leq 1 + a$, *i. e.,* $a(c_1 + 1) \geq ac_2$. When entering l_{i3}, we then have $x = a(c_1 + 1)$, $y = 0$ and $z = a(c_1+1) - ac_2$. Finally, we can go to q_j if $a(c_1+1) \leq 1$ and when entering q_j we have $x = 0$, $y = 1 - a(c_1 + 1)$ and $z = 1 - ac_2$, as expected.

We now examine the second path. We can enter l'_{i2} if $1 - ac_1 \leq a + 1$, *i. e.,* $a(c_1 + 1) \geq 0$, and when entering l'_{i2} we have $x = a(c_1 + 1)$, $y = 0$ and $z = 1 - ac_2 + a(c_1 + 1)$. Then we can go to l_{i3} if $1 - ac_2 + a(c_1 + 1) \leq 1 + a$, *i. e.,* $a(c_1 + 1) \leq ac_2$. When entering l_{i3}, we then have $x = ac_2$, $y = ac_2 - a(c_1 + 1)$ and $z = 0$. Finally, we can go to q_j if $ac_2 \leq 1$ and when entering q_j we have $x = 0$, $y = 1 - a(c_1 + 1)$ and $z = 1 - ac_2$, as expected.

Remark that exactly one path can be taken depending on the respective order of $c_1 + 1$ and c_2, except when both are equal or $a = 0$, in which cases both paths lead to the same configuration anyway (and the case $a = 0$ is excluded by Figure 2a anyway).

Decrement is done similarly by replacing guards $y = a+1$ with $y = 1$, and guards $x = 1$ and $z = 1$ with $x = a+1$ and $z = a + 1$, respectively, as shown in Figure 2c. In addition, the 0-test is obtained by simply adding a transition from q_i to q_k with guard $y = 1 \wedge x = 0$, which ensures that $C_1 = 0$. Similarly, the guard from q_i to l_{i1} ensures that decrement is done only when the counter is not null.

All those gadgets also work for C_2 by swapping y and z.

The actions associated with the transitions do not matter; we can assume a single action σ on all transitions (omitted in all figures).

Finally, we add a self-loop (with no guard) on the location $q_{\mathtt{halt}}$ (encoding the machine state $\mathsf{q}_{\mathsf{halt}}$), ensuring that whenever $q_{\mathtt{halt}}$ is reachable then there exists an infinite run in the PTA.

We now prove that the value of the counters remains bounded iff there exists a parameter valuation v such that $v(\mathcal{A})$ yields an infinite run. First note that if $a = 0$ the initial gadget cannot be passed, and there is no infinite run. Assume $a > 0$. Consider two cases:

1) either the value of the counters is not bounded. Then, for any parameter valuation, at some point during an incrementation of, say, C_1 we will have $a(c_1 + 1) > 1$ when taking the transition from l_{i2} to l_{i3} and the PTA will be blocked. Therefore, there exists no parameter valuation for which there exists an infinite run.
2) or the value of the counters remains bounded. Let c be their maximal value. Let us consider two subcases:
   a) either the machine reaches $\mathsf{q}_{\mathsf{halt}}$: in that case, if $c = 0$ and $0 < a \leq 1$ or $c > 0$ and $ca < 1$, then the PTA valuated with such parameter valuations correctly simulates the machine, yielding a (unique) run reaching location $q_{\mathtt{halt}}$. From there, this run is infinite thanks to the self-loop on $q_{\mathtt{halt}}$. The set of such valuations for a is certainly non-empty: $a = \frac{1}{2}$ belongs to it if $c = 0$ and $a = \frac{1}{c}$ does otherwise.
   b) or the machine does not halt. Then again, for a sufficiently small parameter valuation (*i. e.,* $a < 1$ if $c = 0$ and $a \leq \frac{1}{c}$ otherwise), the machine is properly

simulated, and since the machine does not halt, then the run simulating the infinite execution is infinite too. For other values of a, the machine will block at some point in an increment gadget, because a is not small enough and the guard to q_j cannot be satisfied.

In both subcases, there exist parameter valuations for which there exists an infinite run.

Hence the value of the counters remains bounded iff there exists a parameter valuation v such that $v(\mathcal{A})$ contains an infinite run. □

Remark 1. *In this paper, we allow guards and invariants of the form $x \bowtie \sum_{1 \leq j \leq M} \beta_j p_j + d$, which is more restrictive than [BL09] (that allows parametric coefficients different from 0 and 1, as well as diagonal constraints), but more permissive than [AHV93], that only allows a syntax $x \bowtie p$. In fact, most papers in the literature define their own syntax (see [And15] for a survey). We can adapt our proof to fit in the most restrictive syntax ($x \bowtie p$) as follows: transitions with $y = a + 1$ guards and $y := 0$ reset can be equivalently replaced by one transition with a "$y = 1$" guard and a reset of some additional clock w, followed by a transition with a $w = a$ guard and the $y := 0$ reset (and similarly for x and z is the decrement gadget). This also allows the proof to work without complex parametric expressions in guards, using three additional clocks (we conjecture that a smarter encoding can be exhibited to factor these additional clocks, so as to use a single additional clock). A similar modification can be applied to all subsequent undecidability proofs.*

Finally note that the EC-emptiness problem for the class of open bounded L/U-PTAs (that does not fit in Theorems 1 and 2) remains an open problem. We conjecture that this is decidable using techniques derived from the robustness results of [San11] but the adaptation appears to require rather lengthy developments, with techniques quite different from those presented here, and is thus left to future work.

4. Deadlock-Existence-Emptiness

Theorem 4. *The deadlock-existence-emptiness problem is undecidable for closed bounded L/U-PTAs, with 3 clocks and 2 parameters.*

Proof. We will use a reduction from the halting problem of a 2-counter machine. Let us consider the encoding used in the proof of Theorem 3, that we transform into an L/U-PTA by replacing any comparison of a clock with a (say $x = a$) into $x \leq a^+ \wedge x \geq a^-$, where a^- (resp. a^+) is a lower-bound (resp. upper-bound) parameter. The crux of the proof is in the original enforcement of constraints in the encoding (in particular with location $q'_{\mathtt{halt}}$) such that the deadlock property ensures that $a^- = a^+$.

We give the modified increment gadget in Figure 3 (the decrement gadget is modified in a similar fashion). We replace the initial gadget (Figure 2a) with the new one in Figure 4a. Before initializing the values of the counters, this gadget first ensures that $a^- \leq a^+$.

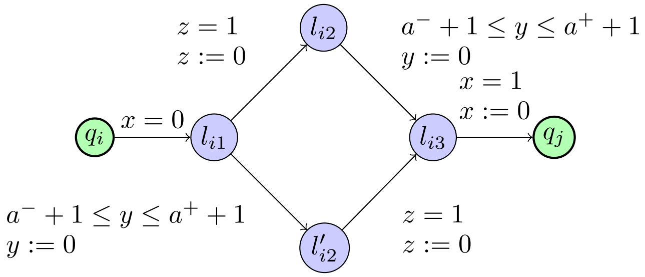
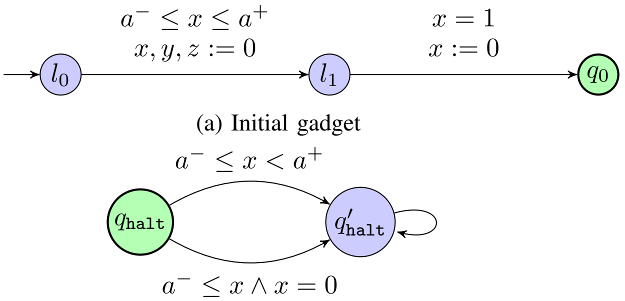


Figure 3: ED-emptiness for bounded L/U-PTAs: increment gadget



(a) Initial gadget

(b) Final gadget

Figure 4: ED-emptiness for bounded L/U-PTAs: initial and final gadgets

We also add a new location $q'_{\mathtt{halt}}$ reachable from $q_{\mathtt{halt}}$ as shown in the final gadget in Figure 4b. Finally, we add an unguarded transition (*i. e.,* a transition the guard of which is true) from any location of the encoding (including that of the initial gadget, but excluding $q_{\mathtt{halt}}$) to location $q'_{\mathtt{halt}}$. That is, it is always possible to reach $q'_{\mathtt{halt}}$ from any location without condition, except from $q_{\mathtt{halt}}$. From that particular location, $q'_{\mathtt{halt}}$ is reachable if and only if $a^- < a^+$ or $a^- = 0$.

We assume the following bounds for the parameters: $a^-, a^+ \in [0, 1]$.

Let us show that there exists a parameter valuation for which the system contains at least one deadlock iff the 2-counter machine halts, which is undecidable [Min67]. Let us reason by cases on the valuations of a^- and a^+.

1) If $a^- > a^+$, the initial gadget cannot be passed, but thanks to the unguarded transitions to $q'_{\mathtt{halt}}$, all runs eventually end in $q'_{\mathtt{halt}}$, from which the absence of deadlock is guaranteed by the unguarded self-loop.
2) If $a^- < a^+$, the machine may not be properly simulated because some transitions do not occur at the right time and some run could reach $q_{\mathtt{halt}}$ while the machine does not halt. Let us consider a run in the TA obtained with such a parameter valuation.
   a) either this run is infinite and remains in the machine (*e. g.,* it loops infinitely through the increment, decrement and 0-test gadgets of our encoding). Then there is no deadlock.
   b) or this run would block in a gadget; in that case, thanks to the unguarded transitions to $q'_{\mathtt{halt}}$, this run

can go to $q'_{\texttt{halt}}$, from which it is deadlock-free.

c) or this run reaches $q_{\texttt{halt}}$ (recall that the value of x is necessarily 0 when entering $q_{\texttt{halt}}$); from there, thanks to the upper transition in Figure 4b, it can reach $q'_{\texttt{halt}}$, from which it is again deadlock-free.

3) If $a^- = a^+ = 0$, the machine may again not be properly simulated: again we could reach $q_{\texttt{halt}}$ while the machine does not halt. The situation is similar to the previous case ($a^- < a^+$) except that in $q_{\texttt{halt}}$ a run has to take the lower transition in Figure 4b to reach $q'_{\texttt{halt}}$, from which it is again deadlock-free.

4) If $a^- = a^+ > 0$:

a) Either the machine does not halt:

i) . . . and the counters remain bounded: for some parameter valuations small enough to encode the value of the counters (typically $a^- = a^+ \leq \frac{1}{c}$, where c is the maximum value of both C_1 and C_2) then the PTA correctly simulates the infinite execution of the machine, and the system is deadlock-free. (Note that such valuations can also lead to $q'_{\texttt{halt}}$ anytime, but this is harmless since this location guarantees the absence of deadlocks.) For other valuations, at some point we have $a^- c_1 > 1$; more specifically, there is an incrementation of C_1 such that $a^- c_1 \leq 1$ and $a^-(c_1 + 1) > 1$. Hence, the run cannot continue in the encoding, but can reach $q'_{\texttt{halt}}$, from where the run is non-blocking.

ii) . . . and the counters are unbounded. Then whatever the value of $a^- > 0$, at some point we have $a^- c_1 > 1$. Then, when executing the corresponding increment gadget, $q'_{\texttt{halt}}$ can be reached from l_{i2}, from where the run is non-blocking.

Hence if the machine does not halt, the system is deadlock-free for all parameter valuations.

b) Or the machine halts. In this case, if c is the maximum value of both C_1 and C_2 over the (necessarily finite) halting execution of the machine, and if $c > 0$, then for valuations such that $a^- = a^+ \leq \frac{1}{c}$, then there exists one run that correctly simulates the machine (beside plenty of runs that will go to $q'_{\texttt{halt}}$ due to the unguarded transitions from all locations except $q_{\texttt{halt}}$); this run that correctly simulates the machine eventually reaches $q_{\texttt{halt}}$. From $q_{\texttt{halt}}$, for such valuations, the system is deadlocked: indeed, the transitions from $q_{\texttt{halt}}$ to $q'_{\texttt{halt}}$ can only be taken if $a^- < a^+$ or $a^- = 0$. And there is no unguarded transition from $q_{\texttt{halt}}$ to $q'_{\texttt{halt}}$, which is crucial for the correctness of our encoding. The set of such valuations for which there exists a run that correctly simulates the machine is certainly non-empty: $a^- = a^+ = \frac{1}{c}$ belongs to it (if $c = 0$ then we choose, *e. g.,* $a^- = a^+ = \frac{1}{2}$). Hence, if the 2-counter machine halts, there exist parameter valuations for which a run has no discrete successor, and hence the system is not deadlock-free.

Hence the 2-counter machine halts iff the set of valuations for which the automaton has at least one deadlock is not empty. □

Corollary 1. *The deadlock-existence-emptiness problem is undecidable for open bounded L/U-PTAs, L/U-PTAs, bounded PTAs and PTAs, with 3 clocks and 2 parameters.*

Proof. Let us consider each formalism:

open bounded L/U-PTAs In the above construction, we can assume, *e. g.,* $a^- \in (0, 1]$, which does not impact the proof.

L/U-PTAs The bounds on the parameters are not required in the above construction: for valuations larger than 1 (that necessarily do not simulate correctly the machine), a gadget may block, therefore leading to $q'_{\texttt{halt}}$, from which the system is deadlock-free, hence without impacting the spirit of the proof.

bounded PTAs From the fact that a bounded L/U-PTA is a bounded PTA.

PTAs From the fact that an L/U-PTA is a PTA.

Observe that the number of parameters can be reduced to 1 for (possibly bounded) PTAs by merging a^- and a^+ into a single parameter a. □

5. EG-Emptiness

In this section, we prove that the EG-emptiness problem is decidable for closed bounded L/U-PTAs, and that lifting either closedness or boundedness leads to undecidability.

Theorem 5. *The EG-emptiness problem is decidable for closed bounded L/U-PTAs.*

We will use Lemma 1 to deal with infinite paths but it is of no use for deadlocks: by decreasing lower-bounds or increasing upper-bounds, some deadlocks can actually be removed. We will therefore also use the symbolic semantics of PTAs (see, *e. g.,* [JLR15]), which we need first to recall.

We define the *time elapsing* of a constraint C, denoted by $C^{\nearrow}$, as the constraint over X and P obtained from C by delaying all clocks by an arbitrary amount of time. That is, $C^{\nearrow} = \{w'|v \mid w \models v(C) \land \forall x \in X : w'(x) = w(x) + d, d \in \mathbb{R}_+\}$. Dually, we define the *past* of C, denoted by $C^{\swarrow}$, as the constraint over X and P obtained from C by letting time pass backward by an arbitrary amount of time. That is, $C^{\swarrow} = \{w'|v \mid w \models v(C) \land \forall x \in X : w'(x) + d = w(x), d \in \mathbb{R}_+\}$. Given $R \subseteq X$, we define the *reset* of C, denoted by $[C]_R$, as the constraint obtained from C by resetting the clocks in R, and keeping the other clocks unchanged. We denote by $C{\downarrow_P}$ the projection of C onto P, *i. e.,* obtained by eliminating the clock variables (*e. g.,* using Fourier-Motzkin).

The (sets of clock valuations satisfying the) constraints generated by PTA can be represented by subsets of $\mathbb{R}_+^{|X|}$ with a special form called *parametric zone* [HRSV02]. A parametric zone is a convex polyhedron over $X \cup P$ in which all constraints on variables are of the form $x \bowtie plt$ (parametric rectangular constraints), or $x_i - x_j \bowtie plt$ (parametric diagonal constraints), where $x_i \in X$, $x_j \in X$ and plt is a parametric linear term over P, *i. e.,* a linear term without clocks ($\alpha_i = 0$ for all i).

A symbolic state is a pair $\mathbf{s} = (l, C)$ where $l \in L$ is a location, and C its associated parametric zone. The initial symbolic state of $\mathcal{A}$ is $\mathbf{s}_0 = \big(l_0, (\{\vec{0}\} \wedge I(l_0))^{\nearrow} \wedge I(l_0)\big)$.

The symbolic semantics relies on the Succ operation. Given a symbolic state $\mathbf{s} = (l, C)$ and an edge $e = (l, g, \sigma, R, l')$, $\mathsf{Succ}(\mathbf{s}, e) = (l', C')$, with $C' = \big([(C \wedge g)]_R \wedge I(l')\big)^{\nearrow} \wedge I(l')$. The Succ operation is effectively computable, using polyhedral operations: note that the successor of a parametric zone C is a parametric zone (see *e. g.,* [JLR15]).

A symbolic run of a PTA is an alternating sequence of symbolic states and edges starting from the initial symbolic state, of the form $\mathbf{s}_0 \overset{e_0}{\Rightarrow} \mathbf{s}_1 \overset{e_1}{\Rightarrow} \cdots \overset{e_{m-1}}{\Rightarrow} \mathbf{s}_m$, such that for all $i = 0, \ldots, m-1$, we have $e_i \in E$, and $\mathbf{s}_{i+1} = \mathsf{Succ}(\mathbf{s}_i, e_i)$. In the following, we simply refer to symbolic states belonging to a run of $\mathcal{A}$ as symbolic states of $\mathcal{A}$.

We can now come back to the proof of Theorem 5.

Proof. Let $\mathcal{A}_{|bounds}$ be a closed bounded L/U-PTA and T be a subset of its locations. Since $\mathcal{A}$ is closed and bounded, for each parameter p, $bounds(p)$ is a closed interval $[m^-(p), m^+(p)]$.

The basic monotonicity property of L/U-PTAs (Lemma 1) ensures that the TA $v_{\mathrm{inf/sup}}(\mathcal{A})$, where $v_{\mathrm{inf/sup}}$ is obtained by valuating lower-bound parameters p^- by $m^-(p)$ and upper-bound parameters p^+ by $m^+(p)$, includes all the runs that could be produced with other parameter valuations. Consequently, if there is an infinite path for some valuation, there is one for $v_{\mathrm{inf/sup}}$ (note that, as emphasized above, this is not true for deadlocks).

In $v_{\mathrm{inf/sup}}(\mathcal{A})$, it is decidable to find an infinite path staying in T, or conclude that none exist: this can be encoded into the CTL formula $EG(T \wedge XG)$, to be verified on the (finite) region graph of $\mathcal{A}$ [AD94]. Since the region equivalence is a time-abstract bisimulation [TY01], this means for $\mathcal{A}$ "there exists a path that remains in T and in which every state has a discrete successor (possibly after letting some time elapse) in T". That path therefore has an infinite number of discrete actions. If we do find such a path, we can then terminate by answering yes to the EG-emptiness problem. If we do not, then in $v_{\mathrm{inf/sup}}(\mathcal{A})$, all paths staying in T are finite. If we keep only discrete actions and locations, which are in finite number, the resulting paths therefore form a finite tree. Let us recall again that, thanks to Lemma 1, all the discrete paths that stay in T and can be obtained with any parameter valuation, belong to that tree.

We can now explicitly compute the symbolic states (following the symbolic semantics recalled above) for all the paths in the finite tree (not only those that are maximal). Recall that each symbolic state $\mathbf{s}$ is a pair (l, C), where l is a location and C a convex polyhedron representing all parameter valuations and clock valuations that can be reached by the given discrete path. In each of these polyhedra, we can explicitly check for the existence of a deadlock: *i*) remove all parts that are in the past of the guard of an outgoing transition in $\mathcal{A}$ (using operation $C^{\swarrow}$), and that would satisfy the target location invariant; *ii*) test for emptiness.

If the result is not empty then there exists a point in the tested set which can be decomposed into a parameter valuation and clock valuation such that, by any time elapsing from the clock valuation, none of the guards can become true. We therefore have a deadlock. If the result is empty, by the same reasoning, we can take a transition (possibly by first letting some time elapse) from all states of C, so none of them are deadlocked. Note that both operations can be performed using classical polyhedral operations.

If we find a deadlock, then we can terminate and answer yes to the EG-emptiness problem. Otherwise, we can terminate and answer no, because we have checked all the potential discrete paths staying in T for any parameter valuation. □

Note that this proof fails when the L/U-PTA is not bounded or closed. In particular, the closedness plays a key role in the sense that we are able to test the valuation $v_{\mathrm{inf/sup}}$. Consider first the L/U-PTA in Figure 1c made of a single location and a single loop with guard $x = 1 \wedge y \leq p$ and a reset of x, where x, y are clocks and p a parameter. This is clearly an L/U-PTA. As p grows, there are more and more discrete behaviors, but there is no cycle for any parameter valuation. In [BL09], the authors provide a finite upper bound $N_{\mathcal{A}}$ for the upper-bound parameters such that if there exists a valuation such that the valuated L/U-PTA has an accepting run, then the valuation giving 0 to lower bound parameters and $N_{\mathcal{A}}$ to upper-bound parameters also ensures the existence of an accepting run. That bound used in this example would indeed prove the non-existence of a cycle for any parameter value, but it does not in turn allow us to derive a finite tree containing all the discrete behaviors, for any possible parameter value (a larger bound would still give more runs).

Similarly, now consider the L/U-PTA in Figure 1d. If 0 is excluded from the domain of p, we have a behavior similar to the previous example: as p gets closer and closer to 0, we have more and more discrete behaviors. And even if we could derive a lower bound à la [BL09] ensuring the non-existence of a cycle here, it would not give a finite tree of all the possible discrete behaviors, for any parameter value.

We can actually exhibit a very thin border between decidability and undecidability of L/U-PTAs by proving that, given a bounded L/U-PTA $\mathcal{A}_{|bounds}$ with a single open bound in *bounds* or an unbounded L/U-PTA, the EG-emptiness problem becomes undecidable.

Theorem 6. *The EG-emptiness problem is undecidable for open bounded L/U-PTAs, with 4 clocks and 4 parameters.*

Proof. We will use a reduction from the halting problem of a 2-counter machine.

Let us consider the encoding used in the proof of Theorem 4, to which we will perform several modifications.

First, we force the 2-counter machine to execute in a constant 1-time unit duration as follows:

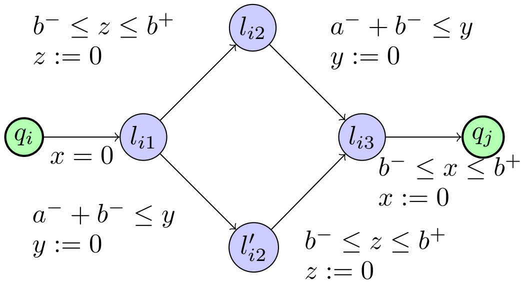


Figure 5: EG-emptiness for bounded L/U-PTAs: increment gadget

1) We replace any occurrence of "1" in the encoding with a parameter, either b^- or b^+ (depending on whether the occurrence of 1 occurs as a lower-bound or an upper-bound); hence the duration of an increment or decrement gadget is now at least b^- and at most b^+. We give the increment gadget in Figure 5. The encoding of a counter is as follows: when $x = 0$, then $y = b - ac_1$ and $z = b - ac_2$, where $a = a^- = a^+$ and $b = b^- = b^+$ (for other parameter valuations, the machine is not properly simulated). Typically, b will need to be sufficiently small compared to 1 to encode the required number of steps of the machine, and a will need to be sufficiently small compared to b to encode the maximum value of the counters. The decrement part of the "test and decrement" instruction is modified similarly.
2) We modify the zero-test part of the "test and decrement" instruction so that its duration is within $[b^-, b^+]$, as in Figure 6: only the first transition encodes the zero-test, the two other transitions forcing $[b^-, b^+]$ time units to elapse while keeping the values of the clocks unchanged, assuming $a^- = a^+$ and $b^- = b^+$ (we will see later that other valuations do not matter). Let $a = a^- = a^+$ and $b = b^- = b^+$. The zero-test requires here that $b = y \wedge x = 0$; in addition, z encodes c_2 as follows: $z = b - ac_2$. After reaching l_{i1} and waiting enough time to take the transition to l_{i2} (*i. e.,* a duration in ac_2) we have: $z = b$ and $x = y = ac_2$. After reaching l_{i2} and waiting enough time to take the transition to q_j (*i. e.,* a duration in $b - ac_2$) we have: $z = b - ac_2$ and $x = y = b$. Resetting x gives $x = 0$, $y = b$ and $z = b - ac_2$, which was the value when performing the 0-test. So the value of the clocks remains unchanged when $b^- = b^+$, and $[b^-, b^+]$ time units have elapsed in any case.
3) We add to any location in the entire system an invariant $w \le 1$, where w is a fresh clock that is never reset in the increment/decrement/zero-test gadgets. (These invariants are omitted in Figure 5.)

Hence, the duration of any gadget is at least b^- and therefore for any valuation $b^- > 0$ the number of operations the machine can perform is finite due to the global invariant $w \le 1$.

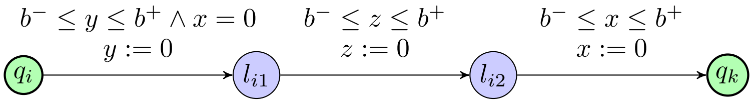


Figure 6: EG-emptiness for bounded L/U-PTAs: zero-test gadget

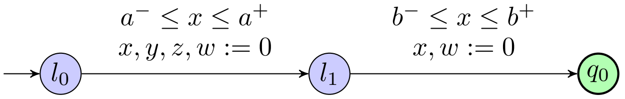


Figure 7: EG-emptiness for bounded L/U-PTAs: initial gadget

Then, before starting the 2-counter machine encoding, we add an initial gadget given in Figure 7. This gadget constrains $a^- \le a^+$, $b^- \le b^+$, and is such that when leaving the gadget then $y, z \in [b^- \le b^+]$ while x, w are 0. When $b^- = b^+$, this correctly encodes that the value of both counters is 0.

Then, we add a new $q'_{\mathtt{halt}}$ location (without any invariant, *i. e.,* not requiring $w \le 1$), with two transitions from $q_{\mathtt{halt}}$ as depicted in Figure 8. We then add a transition (with no guard) from any location of the encoding (except $q_{\mathtt{halt}}$) to $q'_{\mathtt{halt}}$. That is, for any increment gadget, if the value of the parameters is not small enough to correctly simulate the machine, then the system is not deadlocked, and can lead instead to $q'_{\mathtt{halt}}$. (If the value is small enough, the system can either lead to $q'_{\mathtt{halt}}$ or continue in the 2-counter machine encoding.) We also add a transition to $q'_{\mathtt{halt}}$ (with no guard) from all locations in the initial gadget in Figure 7.

We assume the following bounds for the parameters: $a^-, a^+, b^+ \in [0, 1]$ and $b^- \in (0, 1]$.

Let us show that the 2-counter machine halts iff the set of valuations satisfying $\text{EG}(L \setminus \{q'_{\mathtt{halt}}\})$ is not empty.

1) If $a^- > a^+$ or $b^- > b^+$, the initial gadget cannot be passed, and thanks to the transitions to $q'_{\mathtt{halt}}$, all runs eventually reach $q'_{\mathtt{halt}}$, hence $\text{EG}(L \setminus \{q'_{\mathtt{halt}}\})$ does not hold.
2) If $a^- < a^+$ and $b^- \le b^+$, then the machine may not be correctly simulated: a given run will either reach $q_{\mathtt{halt}}$, in which case it will also reach $q'_{\mathtt{halt}}$ (as the guard from $q_{\mathtt{halt}}$ to $q'_{\mathtt{halt}}$ does not forbid this run), or it will loop in the machine until it eventually gets blocked (since $b^- > 0$ and because of the invariant $w \le 1$, for any value of b^-, the maximal number of steps is $\frac{1}{b^-}$); when being blocked, it has no other option than going to $q'_{\mathtt{halt}}$, thanks to the unguarded transitions from any

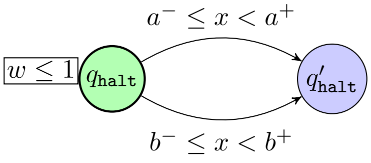


Figure 8: EG-emptiness for bounded L/U-PTAs: final gadget

location to $q'_{\mathtt{halt}}$. Hence if $a^- < a^+$, $\mathrm{EG}(L \setminus \{q'_{\mathtt{halt}}\})$ does not hold.

3) If $b^- < b^+$ (and $a^- \leq a^+$), again the machine may not be correctly simulated, and following a similar reasoning, $\mathrm{EG}(L \setminus \{q'_{\mathtt{halt}}\})$ again does not hold.
4) If $a^- = a^+$ and $b^- = b^+ > 0$:
   a) Either the machine does not halt: in this case, after a maximum number of steps (typically $\frac{1}{b^-}$), a gadget will be blocked due to the invariant $w \leq 1$, and the run will end in $q'_{\mathtt{halt}}$. Hence if the 2-counter machine does not halt, $\mathrm{EG}(L \setminus \{q'_{\mathtt{halt}}\})$ does not hold.
   b) Or the machine halts: in this case, if c is the maximum value of both C_1 and C_2 over the (necessarily finite) halting execution of the machine, and if m is the length of this execution, and if $c > 0$, then for valuations such that $a^- = a^+ \leq \frac{b^-}{c}$ and $b^- = b^+ \leq \frac{1}{m}$, then there exists one run that correctly simulates the machine (beside plenty of runs that will go to $q'_{\mathtt{halt}}$ due to the unguarded transitions); this run that correctly simulates the machine eventually reaches $q_{\mathtt{halt}}$. From $q_{\mathtt{halt}}$, for such valuations, the system is deadlocked: indeed, the transitions from $q_{\mathtt{halt}}$ to $q'_{\mathtt{halt}}$ can only be taken if $a^- < a^+$ or $b^- < b^+$. Hence $\mathrm{EG}(L \setminus \{q'_{\mathtt{halt}}\})$ holds. The set of such valuations is certainly non-empty: $a^- = a^+ = \frac{1}{m \times c}$ and $b^- = b^+ = \frac{1}{m}$ belongs to it (if $c = 0$ then we choose, *e. g.*, $b^- = b^+ = 1$ and $a^- = a^+ = \frac{1}{2}$). Hence, if the 2-counter machine halts, there exist parameter valuations for which $\mathrm{EG}(L \setminus \{q'_{\mathtt{halt}}\})$ holds.

Hence the 2-counter machine halts iff the set of valuations for which $\mathrm{EG}(L \setminus \{q'_{\mathtt{halt}}\})$ holds is not empty. □

Remark 2. *The above construction works over 1 time unit (an invariant can be added to $q'_{\mathtt{halt}}$ too), so this gives an undecidability result over bounded time as well.*

We now prove that EG-emptiness is also undecidable for unbounded L/U-PTAs. When not considering L/U-PTAs, proving an undecidability result for bounded PTAs gives the undecidability for unbounded PTAs, as a bounded PTA can be simulated using a PTA (by, *e. g.*, adding the bounds as a guard between a fresh location prior to the initial location and the initial location, *e. g.*, $p \in [\inf, \sup]$ becomes $\inf \leq x \leq \sup \wedge p = x$). Recall that this is not true for L/U-PTAs, as such a construction requires to compare the clock and the parameter using an equality; in addition, L/U-PTAs are incomparable with bounded L/U-PTAs [ALR16b]. In addition, our proof for unbounded L/U-PTAs uses one parameter less than for open bounded L/U-PTAs.

Theorem 7. *The EG-emptiness problem is undecidable for L/U-PTAs with 4 clocks and 3 parameters.*

Proof (sketch). We again use a reduction from the halting problem of a 2-counter machine. Our proof essentially relies on a mechanism similar to the proof of Theorem 6. However, we must use a different PTA encoding (the encoding used in the proof of Theorem 6 does not work for unbounded L/U-PTAs, as it strongly relies on the fact that b^- be strictly positive). Instead, we propose an encoding inspired by that of a 2-counter machine proposed in [BBLS15] to prove the undecidability of the EF-emptiness problem for PTAs with a single integer-valued parameter (that can also be rational-valued). We modify the encoding of [BBLS15] to obtain an L/U-PTA, by splitting the single parameter a into a lower-bound parameter a^- and an upper-bound parameter a^+, in the spirit of previous undecidability results for L/U-PTAs in this paper (Theorems 4 and 6). Then, we add a global invariant $w \leq b^+$ (where w is a fresh clock never reset, and b^+ a fresh upper-bound parameter), to ensure that, for any valuation of $b^+ > 0$, the number of operations the machine can perform is finite (which requires some modifications of the gadgets to ensure that they require at least 1 time unit). The proof then follows a reasoning similar to that of Theorem 6.

See Section 7.1 for a detailed proof. □

Remark 3. *The above construction works also for integer-valued parameters, so this gives an undecidability result for integer-valued parameters too. The proof also works over discrete time (with integer-valued parameters).*

6. Conclusion

Despite the vast number of undecidability results linked to the formalism of parametric timed automata, and to which we also contribute here, we have achieved some decidability for the existential parametric problem on the EG liveness property. This could be done by imposing original constraints to the classical subclass of L/U-PTAs, pertaining to the topology of the domain of the parameter values. This domain should be a closed and bounded hyperrectangle of the rational space.

The subclass together with the EG property really lies on the boundary of decidability: on the one hand, we have proved that considering unbounded, or bounded but open domains leads again to undecidability for EG. On the other hand, if we consider — instead of the EG property which asks for the existence of a maximal finite or infinite path staying in some locations — only infinite maximal paths (existence of discrete cycles), then we have proved that the problem becomes decidable (for either closed bounded domains, or unbounded domains — the case of open bounded domains remains open). And finally, if we consider only finite maximal paths (existence of deadlocks), then we have proved that the problem becomes consistently undecidable. Our results are summarized in Table 1, where bold green denotes decidability and italic red denotes undecidability.

Future work includes *i*) studying the decidability of EC-emptiness for open bounded L/U-PTAs (possibly adapting techniques developed in [San11]), *ii*) extending the EG decidability result to shapes other than hyperrectangles, and *iii*) studying actual synthesis. In addition, the decidability of problems we proved undecidable for L/U-PTAs should be studied for two subclasses of L/U-PTAs, where all parameters are upper bounds (U-PTAs) or all lower bounds (L-PTAs).

Class	PTAs	L/U-PTAs	bounded open L/U-PTAs	bounded closed L/U-PTAs
EC-emptiness	*Theorem 3*	**Theorem 1**	open	**Theorem 2**
ED-emptiness	*Corollary 1*	*Corollary 1*	*Theorem 4*	*Corollary 1*
EG-emptiness	*from Theorem 7*	*Theorem 7*	*Theorem 6*	**Theorem 5**

TABLE 1: Decidability of EG for L/U-PTAs

Acknowledgments

The authors thank Olivier H. Roux for fruitful discussions on the topic of parametric timed automata, as well as some reviewers for useful comments.

References

[AD94] Rajeev Alur and David L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126(2):183–235, 1994.

[AHV93] Rajeev Alur, Thomas A. Henzinger, and Moshe Y. Vardi. Parametric real-time reasoning. In *STOC*, pages 592–601. ACM, 1993.

[ALR16a] Étienne André, Didier Lime, and Olivier H. Roux. Decision problems for parametric timed automata. In *ICFEM*, volume 10009 of *Lecture Notes in Computer Science*, pages 400–416. Springer, 2016.

[ALR16b] Étienne André, Didier Lime, and Olivier H. Roux. On the expressiveness of parametric timed automata. In *FORMATS*, volume 9984 of *Lecture Notes in Computer Science*, pages 19–34. Springer, 2016.

[And15] Étienne André. What's decidable about parametric timed automata? In *FTSCS*, volume 596 of *Communications in Computer and Information Science*, pages 1–17. Springer, 2015.

[BBLS15] Nikola Beneš, Peter Bezděk, Kim G. Larsen, and Jiří Srba. Language emptiness of continuous-time parametric timed automata. In *ICALP, Part II*, volume 9135 of *Lecture Notes in Computer Science*, pages 69–81. Springer, 2015.

[BL09] Laura Bozzelli and Salvatore La Torre. Decision problems for lower/upper bound parametric timed automata. *Formal Methods in System Design*, 35(2):121–151, 2009.

[BO14] Daniel Bundala and Joël Ouaknine. Advances in parametric real-time reasoning. In *MFCS*, volume 8634 of *Lecture Notes in Computer Science*, pages 123–134. Springer, 2014.

[CES86] Edmund M. Clarke, E. Allen Emerson, and A. Prasad Sistla. Automatic verification of finite-state concurrent systems using temporal logic specifications. *ACM Transactions on Programming Languages and Systems*, 8(2):244–263, 1986.

[Doy07] Laurent Doyen. Robust parametric reachability for timed automata. *Information Processing Letters*, 102(5):208–213, 2007.

[HRSV02] Thomas Hune, Judi Romijn, Mariëlle Stoelinga, and Frits W. Vaandrager. Linear parametric model checking of timed automata. *Journal of Logic and Algebraic Programming*, 52-53:183–220, 2002.

[JLR15] Aleksandra Jovanović, Didier Lime, and Olivier H. Roux. Integer parameter synthesis for timed automata. *IEEE Transactions on Software Engineering*, 41(5):445–461, 2015.

[Lam77] Leslie Lamport. Proving the correctness of multiprocess programs. *IEEE Transactions on Software Engineering*, 3(2):125–143, 1977.

[Mil00] Joseph S. Miller. Decidability and complexity results for timed automata and semi-linear hybrid automata. In *HSCC*, volume 1790 of *Lecture Notes in Computer Science*, pages 296–309. Springer, 2000.

[Min67] Marvin L. Minsky. *Computation: finite and infinite machines*. Prentice-Hall, Inc., 1967.

[San11] Ocan Sankur. Untimed language preservation in timed systems. In *MFCS*, volume 6907 of *Lecture Notes in Computer Science*, pages 556–567. Springer, 2011.

[TY01] Stavros Tripakis and Sergio Yovine. Analysis of timed systems using time-abstracting bisimulations. *Formal Methods in System Design*, 18(1):25–68, 2001.

7. Appendix

7.1. Proof of Theorem 7

Theorem 7 (recalled). *The EG-emptiness problem is undecidable for L/U-PTAs with 4 clocks and 3 parameters.*

Proof. We will again use a reduction from the halting problem of a 2-counter machine. Our proof essentially relies on a mechanism similar to the proof of Theorem 6; however, we must use a different PTA encoding (the encoding used in the proof of Theorem 6 does not work for unbounded L/U-PTAs, as it strongly relies on the fact that b^- be strictly positive), which prevents us to factor the proof as much as we would have wished.

We propose here an encoding inspired by that of a 2-counter machine proposed in [BBLS15] to prove the undecidability of the EF-emptiness problem for PTAs with a single integer-valued parameter used to encode the maximum value of the two counters (although not considered in [BBLS15], the proof also works identically with a rational-valued parameter). Two different instructions are considered:

- when in state q_i, increment C_k and go to q_j;
- when in state q_i, if $C_k = 0$ then go to q_k, otherwise decrement C_k and go to q_j;

Starting from the initial configuration $(\mathsf{q}_0, C_1 = 0, C_2 = 0)$ the machine either reaches $q_{\mathtt{halt}}$ and halts, or loops forever. Knowing whether the machine halts is undecidable [Min67].

The encoding uses a single parameter a. Two clocks x and y are used to encode the value of the counters, while a third clock z is used as an auxiliary clock. Whenever $z = 0$, then $x = c_1$ and $y = c_2$.

We modify this encoding by splitting the single parameter a into a lower-bound parameter a^- and an upper-bound parameter a^+, in the spirit of previous undecidability results for L/U-PTAs in this paper (Theorems 4 and 6).

In addition, we request that the entire execution takes a time less than b^+, where b^+ is a fresh upper-bound parameter; this is achieved by adding an invariant $w \leq b^+$ to all locations (with w a fresh clock never reset after the initial gadget).

We give the modified increment gadget for the first counter in Figure 9 (invariants are omitted). Note that, if $z = 0$ when entering q_i then the time to pass this gadget is in $[a^- + 1, a^+ + 1]$.

The test and decrement gadget is similar, and given in Figure 10. We performed a slight modification to the zero-test of [BBLS15], that was executed in 0-time; we require in our construction that each gadget takes at least one time unit. Hence, we rewrote it in Figure 10 so as to force at least one time unit to elapse after the clocks are tested, and so that the final value of the clock is not changed, when $a^- = a^+$ (in the spirit of the same operation in the proof of Theorem 6): when performing the zero-test, we have $x = z = 0$ and $y = c_2$. Then after $a - c_2 + 1$ time units (with $a = a^+ = a^-$), we have $x = z = a + 1 - c_2$ and $y = a + 1$, and we can take the transition to $l_{i2''}$, resetting y. Then after c_2 time units, we have $x = z = a + 1$ and $y = c_2$ and we can take the transition to $l_{i2''}$, resetting x and z. This gives finally $x = z = 0$ and $y = c_2$ and the time spent in the gadget is in $[a^- + 1, a^+ + 1]$, and therefore is more than one time unit. Gadgets for the second counter are symmetric.

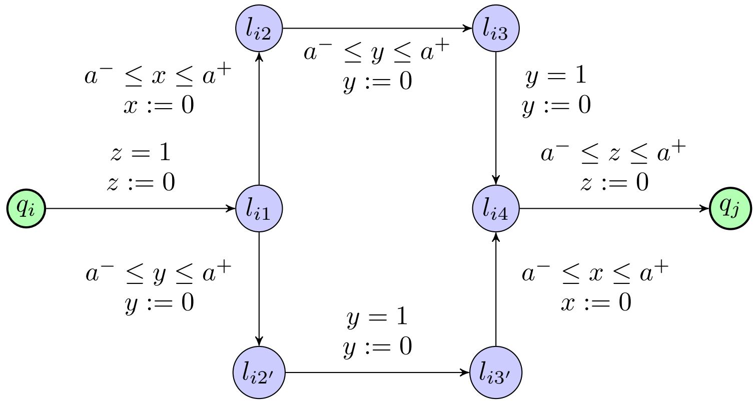


Figure 9: EG-emptiness for L/U-PTAs: increment gadget

We add before the first instruction the initial gadget given in Figure 11, constraining $a^- \leq a^+$ and $b^+ > 0$, and resetting all clocks.

In addition, just as in Theorem 6, we add unguarded transitions from any location (including that of the initial gadget, but excluding $q_{\mathtt{halt}}$) to a new location $q'_{\mathtt{halt}}$. We also add two transitions from $q_{\mathtt{halt}}$ to $q'_{\mathtt{halt}}$ given in the final gadget in Figure 12.

Let us show that the 2-counter machine halts iff the set of valuations for which $\mathrm{EG}(L \setminus \{q'_{\mathtt{halt}}\})$ holds is not empty. We reason on the parameter valuations.

1) If $a^- > a^+$ or $b^+ = 0$, the initial gadget cannot be passed: any run is sent to $q'_{\mathtt{halt}}$ because of the transitions to $q'_{\mathtt{halt}}$, and therefore $\mathrm{EG}(L \setminus \{q'_{\mathtt{halt}}\})$ does not hold.
2) If $a^- < a^+$ and $b^+ > 0$, then the machine may not be correctly simulated: a given run will either reach $q_{\mathtt{halt}}$, in which case it will also reach $q'_{\mathtt{halt}}$ (as the guard from $q_{\mathtt{halt}}$ to $q'_{\mathtt{halt}}$ in Figure 12 does not forbid this run), or it will loop in the machine until it eventually gets blocked: since $b^+ > 0$, since all gadgets require at least 1 time unit, for any value of b^+ the invariant $z \leq b^+$ will eventually block a transition after at most b^+ steps. When being blocked, a run has no other option than going to $q'_{\mathtt{halt}}$, because of the unguarded transitions from any location to $q'_{\mathtt{halt}}$. Hence if $a^- < a^+$ and $b^+ > 0$, $\mathrm{EG}(L \setminus \{q'_{\mathtt{halt}}\})$ does not hold.
3) Now, assume $a^- = a^+$ and $b^+ > 0$.
   a) Either the machine does not halt: in this case, after a maximum number of steps (typically at most b^+), a gadget will be blocked due to the invariant $z \leq b^+$, and the run will end in $q'_{\mathtt{halt}}$ because of the unguarded transitions from any location to $q'_{\mathtt{halt}}$. Hence if the 2-counter machine does not halt, $\mathrm{EG}(L \setminus \{q'_{\mathtt{halt}}\})$ does not hold.

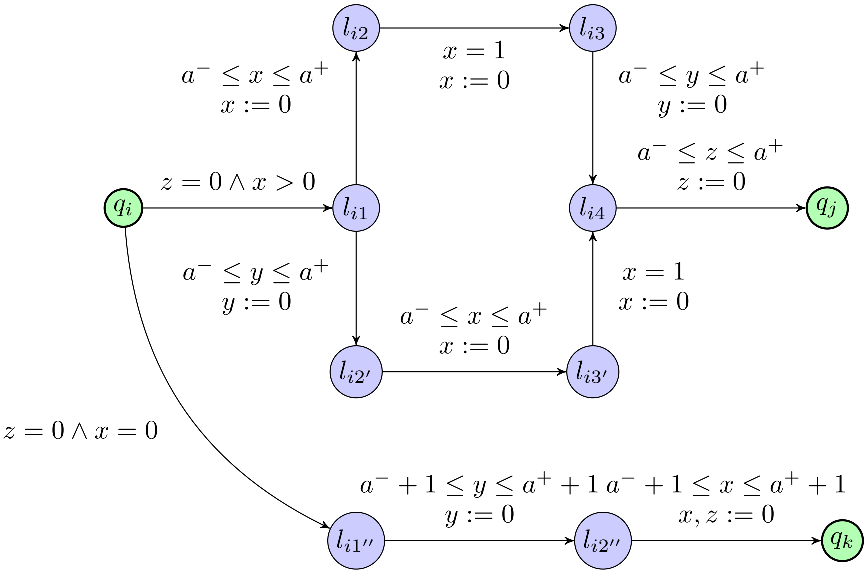


Figure 10: EG-emptiness for L/U-PTAs: test and decrement gadget

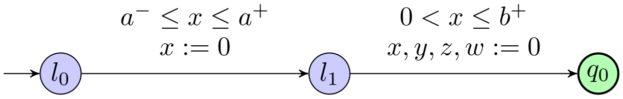


Figure 11: EG-emptiness for L/U-PTAs: initial gadget

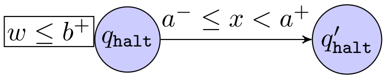


Figure 12: EG-emptiness for L/U-PTAs: final gadget

b) Or the machine halts: in this case, if c is the maximum value of both C_1 and C_2 over the (necessarily finite) halting execution of the machine, and if m is the length of this execution, and if $c > 0$, then for valuations such that $a^- = a^+ \leq c$ and sufficiently large valuations of b^+ (typically $b^+ \geq m \times (a^+ + 1)$ as a gadget can take up to $a^+ + 1$ time units), then there exists one run that correctly simulates the machine; this run eventually reaches $q_{\texttt{halt}}$. From $q_{\texttt{halt}}$, for such values, the system is deadlocked. Hence, if the 2-counter machine halts, there exist parameter valuations for which a run does not reach $q'_{\texttt{halt}}$, *i. e.,* for which $\text{EG}(L \setminus \{q'_{\texttt{halt}}\})$ holds.

Hence the 2-counter machine halts iff the set of valuations for which $\text{EG}(L \setminus \{q'_{\texttt{halt}}\})$ holds is not empty. □