



**HAL**  
open science

## Liveness in L/U-Parametric Timed Automata

Étienne André, Didier Lime

► **To cite this version:**

| Étienne André, Didier Lime. Liveness in L/U-Parametric Timed Automata. 2016. hal-01304232

**HAL Id: hal-01304232**

**<https://hal.science/hal-01304232v1>**

Preprint submitted on 19 Apr 2016

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

# Liveness in L/U-Parametric Timed Automata\*

Étienne André and Didier Lime

École Centrale de Nantes, IRCCyN, CNRS, UMR 6597, 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 existence of a valuation for which a run remains in a given set of locations exhibits a very thin border between decidability and undecidability.

**1998 ACM Subject Classification** D.2.4 Software/Program Verification: Formal methods

**Keywords and phrases** L/U-PTA, EG-emptiness, deadlock-freeness, infinite run

**Digital Object Identifier** 10.4230/LIPIcs.CVIT.1942.23

## 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”) [11]. 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 [7].

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.

Parametric timed automata (PTAs) [2] 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.*, [12, 8, 5, 3]).

---

\* This work is partially supported by the ANR national research program “PACS” (ANR-14-CE28-0002).



1 In order to overcome these disappointing results, lower-bound/upper-bound parametric  
 2 timed automata (L/U-PTAs) are introduced as a subclass of PTAs where each parameter  
 3 either always appears as an upper bound when compared to a clock, or always as a lower  
 4 bound [9]. The EF-emptiness problem, and also the EF-universality problem (“Can we reach  
 5 a given location, regardless of what valuations we give to the parameters?”) are decidable  
 6 for L/U-PTAs.

7 In [6], infinite acceptance properties are considered: the emptiness and the universality  
 8 of the valuation set for which a given location is infinitely often traversed are decidable for  
 9 integer-valued parameters. In [10], it is shown that the AF-emptiness problem (“Does there  
 10 exist a valuation of the parameters, such that the system reaches a given location for all  
 11 runs?”) is undecidable for L/U-PTAs with integer- and rational-valued parameters.

## 12 Contribution

13 With the notable exception of [10], and to some extent of [6] which addresses the existence  
 14 of cycles, all the works cited above focus on safety properties, through the basic problem of  
 15 reachability. This is maybe not so surprising given that most results related to this simpler  
 16 problem are already negative.

17 We nonetheless address here the problem of liveness in PTAs, and more precisely, with  
 18 the negative result of [10] on AF-emptiness in mind, we start from L/U-PTAs with rational-  
 19 valued parameters and further refine both the model and the properties. We prove that:

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

## 30 Outline

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

## 35 **2** Preliminaries

### 36 2.1 Clocks, Parameters and Constraints

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

41 Throughout this paper, we assume a set  $X = \{x_1, \dots, x_H\}$  of *clocks*, *i. e.*, real-valued  
 42 variables that evolve at the same rate. A clock valuation is a function  $w : X \rightarrow \mathbb{R}_+$ . We

1 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  
 2 clock valuation that assigns 0 to all clocks. Given  $d \in \mathbb{R}_+$ ,  $w + d$  denotes the valuation such  
 3 that  $(w + d)(x) = w(x) + d$ , for all  $x \in X$ . Given  $R \subseteq X$ , we define the *reset* of a valuation  $w$ ,  
 4 denoted by  $[w]_R$ , as follows:  $[w]_R(x) = 0$  if  $x \in R$ , and  $[w]_R(x) = w(x)$  otherwise.

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

8 In the following, we assume  $\sim \in \{<, \leq, \geq, >\}$ . Throughout this paper,  $lt$  denotes a linear  
 9 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  
 10  $\alpha_i, \beta_j, d \in \mathbb{Z}$ . A *constraint*  $C$  (*i. e.*, a convex polyhedron) over  $X \cup P$  is a conjunction of  
 11 inequalities of the form  $lt \sim 0$ . Given a parameter valuation  $v$ ,  $v(C)$  denotes the constraint  
 12 over  $X$  obtained by replacing each parameter  $p$  in  $C$  with  $v(p)$ . Likewise, given a clock  
 13 valuation  $w$ ,  $w(v(C))$  denotes the expression obtained by replacing each clock  $x$  in  $v(C)$   
 14 with  $w(x)$ . We say that  $v$  *satisfies*  $C$ , denoted by  $v \models C$ , if the set of clock valuations  
 15 satisfying  $v(C)$  is nonempty. Given a parameter valuation  $v$  and a clock valuation  $w$ , we  
 16 denote by  $w|v$  the valuation over  $X \cup P$  such that for all clocks  $x$ ,  $w|v(x) = w(x)$  and for  
 17 all parameters  $p$ ,  $w|v(p) = v(p)$ . We use the notation  $w|v \models C$  to indicate that  $w(v(C))$   
 18 evaluates to true. We say that  $C$  is *satisfiable* if  $\exists w, v$  s.t.  $w|v \models C$ .

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

## 2.2 Parametric Timed Automata

21 **► Definition 1.** A PTA  $\mathcal{A}$  is a tuple  $\mathcal{A} = (\Sigma, L, l_0, X, P, I, E)$ , where: *i*)  $\Sigma$  is a finite set of  
 22 actions, *ii*)  $L$  is a finite set of locations, *iii*)  $l_0 \in L$  is the initial location, *iv*)  $X$  is a finite set  
 23 of clocks, *v*)  $P$  is a finite set of parameters, *vi*)  $I$  is the invariant, assigning to every  $l \in L$  a  
 24 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  
 25 target locations,  $\sigma \in \Sigma$ ,  $R \subseteq X$  is a set of clocks to be reset, and  $g$  is a guard.

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

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

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

36 Moreover we write  $(l, w) \xrightarrow{e} (l', w')$  for a sequence of delay and discrete transitions where  
 37  $((l, w), e, (l', w')) \in \rightarrow$  if  $\exists d, w'' : (l, w) \xrightarrow{d} (l, w'') \xrightarrow{e} (l', w')$ .

38 Given a TA  $v(\mathcal{A})$  with concrete semantics  $(S, s_0, \rightarrow)$ , we refer to the states of  $S$  as the  
 39 *concrete states* of  $v(\mathcal{A})$ . A *concrete run* (or simply a *run*) of  $v(\mathcal{A})$  is an alternating sequence  
 40 of concrete states of  $v(\mathcal{A})$  and edges starting from the initial concrete state  $s_0$  of the form  
 41  $s_0 \xrightarrow{e_0} s_1 \xrightarrow{e_1} \dots \xrightarrow{e_{m-1}} s_m$ , such that for all  $i = 0, \dots, m - 1$ ,  $e_i \in E$ , and  $(s_i, e_i, s_{i+1}) \in \rightarrow$ .  
 42 Given a state  $s = (l, w)$ , we say that  $s$  is *reachable* (or that  $v(\mathcal{A})$  reaches  $s$ ) if  $s$  belongs  
 43 to a run of  $v(\mathcal{A})$ . By extension, we say that  $l$  is *reachable* in  $v(\mathcal{A})$ , if there exists a state  
 44

1  $(l, w)$  that is reachable. Given a set of locations  $G \subseteq L$ , we say that a run stays in  $G$  if  
 2 all of its states  $(l, w)$  are such that  $l \in G$ . A maximal run is a run that is either infinite  
 3 (*i. e.*, contains an infinite number of discrete transitions), or that cannot be extended by a  
 4 discrete transition. A maximal run is deadlocked if it is finite, *i. e.*, contains a finite number  
 5 of discrete transitions. By extension, we say that a TA is deadlocked if it contains at least  
 6 one deadlocked run.

## 7 2.3 Subclasses of PTAs

8 ► **Definition 3** (L/U-PTA). An L/U-PTA is a PTA where the set of parameters is partitioned  
 9 into lower-bound parameters and upper-bound parameters, where an upper-bound (resp.  
 10 lower-bound) parameter  $p_i$  is such that, whenever it appears in a guard or an invariant  
 11  $x \sim \sum_{1 \leq j \leq M} \beta_j p_j + d$  (where  $\beta_i = 1$ ) then necessarily  $\sim \in \{\leq, <\}$  (resp.  $\sim \in \{\geq, >\}$ ).

12 L/U-PTAs enjoy a well-known monotonicity property recalled in the following lemma  
 13 (that corresponds to a reformulation of [9, Prop 4.2]), stating that increasing upper-bound  
 14 parameters or decreasing lower-bound parameters can only add behaviors.

15 ► **Lemma 4.** Let  $\mathcal{A}$  be an L/U-PTA and  $v$  be a parameter valuation. Let  $v'$  be a valuation  
 16 such that for each upper-bound parameter  $p^+$ ,  $v'(p^+) \geq v(p^+)$  and for each lower-bound  
 17 parameter  $p^-$ ,  $v'(p^-) \leq v(p^-)$ . Then any run of  $v(\mathcal{A})$  is a run of  $v'(\mathcal{A})$ .

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

20 ► **Definition 5** (bounded PTA). A *bounded PTA* is  $\mathcal{A}|_{\text{bounds}}$ , where  $\mathcal{A}$  is a PTA, and  $\text{bounds} : P \rightarrow \mathcal{I}(\mathbb{N})$   
 21 assigns to each parameter  $p$  an interval  $[\text{inf}, \text{sup}]$ ,  $(\text{inf}, \text{sup}]$ ,  $[\text{inf}, \text{sup})$ , or  $(\text{inf}, \text{sup})$ ,  
 22 with  $\text{inf}, \text{sup} \in \mathbb{N}$ . We use  $\text{inf}(p, \text{bounds})$  and  $\text{sup}(p, \text{bounds})$  to denote the infimum and the  
 23 supremum of  $p$ , respectively. (Note that we rule out  $\infty$  as a supremum.)

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

26 We define similarly bounded L/U-PTAs.

## 27 2.4 Decision Problems

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

### **$\mathcal{P}$ -emptiness problem:**

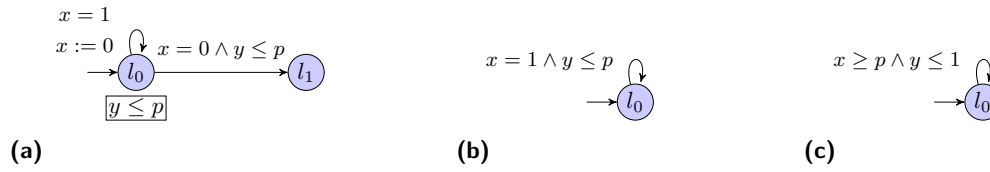
29 INPUT: A PTA  $\mathcal{A}$  and an instance  $\phi$  of  $\mathcal{P}$

PROBLEM: Is the set of parameter valuations  $v$  such that  $v(\mathcal{A})$  satisfies  $\phi$  empty?

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

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

37 For example, given a PTA  $\mathcal{A}$ , deadlock-existence-emptiness asks: “does there exist a  
 38 valuation  $v$  of the parameters such that at least one run of  $v(\mathcal{A})$  is deadlocked, *i. e.*, has no  
 39 discrete successor”. In the following, we often abbreviate deadlock-existence-emptiness and  
 40 cycle-existence-emptiness as ED-emptiness and EC-emptiness, respectively.



■ **Figure 1** Examples of L/U-PTAs

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

### 3 Cycle-Existence-Emptiness

8 ► **Theorem 6.** *The cycle-existence-emptiness problem is decidable for closed bounded L/U-*  
 9 *PTAs.*

10 **Proof (sketch).** From Lemma 4, this problem is equivalent to testing the TA obtained by  
 11 valuating upper-bound (resp. lower-bound) parameters with their maximal (resp. minimum)  
 12 value in *bounds*. See Appendix A for a detailed proof. ◀

13 The above result cannot be used as such for non-bounded L/U-PTAs as a cycle that  
 14 exists for an infinite parameter valuation may not exist for any finite parameter valuation:  
 15 consider the L/U-PTA in Figure 1a. This L/U-PTA has an infinite run for  $p = \infty$ , but for  
 16 any parameter valuation (*i. e.*, different from  $\infty$ ), the number of self-loops in  $l_0$  is bounded  
 17 by  $p$ , and hence finite. However, extending to rational-valued parameters a result from [6],  
 18 we can still prove decidability.

19 ► **Lemma 7.** *Given an L/U-PTA  $\mathcal{A}$  and a subset of its locations  $G$ , the problem of the*  
 20 *existence of at least one parameter valuation  $v$  such that  $v(\mathcal{A})$  has a run passing infinitely*  
 21 *often through  $G$  is PSPACE-complete.*

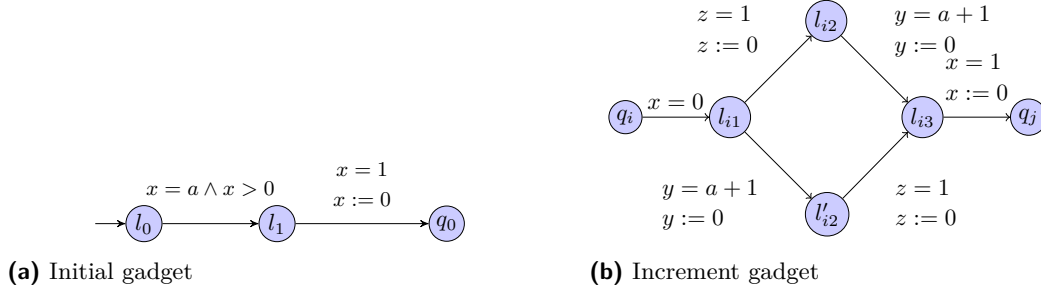
22 **Proof (sketch).** We show that there exists a rational-valued valuation yielding such an  
 23 infinite run iff there exists an integer-valued valuation yielding such an infinite run. We can  
 24 then apply [6, Theorem 8], that proves that the existence of such a valuation is PSPACE-  
 25 complete. See Appendix B for a detailed proof. ◀

26 ► **Theorem 8.** *The cycle-existence-emptiness problem is PSPACE-complete for L/U-PTAs.*

27 **Proof.** Let  $\mathcal{A}$  be an L/U-PTA. The set of parameter valuations for which  $\mathcal{A}$  has an infinite  
 28 run is empty iff the set of parameter valuations for which  $\mathcal{A}$  has an infinite run passing  
 29 infinitely often through  $L$  (where  $L$  denotes all locations of  $\mathcal{A}$ ) is empty. Hence we can  
 30 directly apply our intermediate Lemma 7 to conclude that this problem is decidable and  
 31 PSPACE-complete. ◀

32 Without surprise, this problem becomes undecidable for general PTAs, even when bounded.

33 ► **Theorem 9.** *The cycle-existence-emptiness problem is undecidable for (bounded) PTAs*  
 34 *with 3 clocks and 1 parameter.*



■ **Figure 2** EC-emptiness: gadgets

1 **Proof.** We reduce from the boundedness problem of a 2-counter machine, which is undecid-  
 2 able [13]. Recall that a deterministic 2-counter machine has two non-negative counters  $C_1$   
 3 and  $C_2$ , a finite number of states and a finite number of transitions, which can be of the  
 4 form:

- 5 ■ when in state  $q_i$ , increment  $C_k$  and go to  $q_j$ ;
- 6 ■ when in state  $q_i$ , decrement  $C_k$  and go to  $q_j$ ;
- 7 ■ when in state  $q_i$ , if  $C_k = 0$  then go to  $q_j$ , otherwise block.

8 The machine starts in state  $q_0$ ; by definition, it halts when it reaches a specific state  
 9 called  $q_{\text{halt}}$ . The boundedness problem for 2-counter machines asks whether the value of the  
 10 counters remains smaller than some bound, and is undecidable [13].

11 Given such a machine  $\mathcal{M}$ , we encode it as a PTA  $\mathcal{A}(\mathcal{M})$ ; our encoding is adapted from  
 12 an existing encoding of a 2-counter machine, used to (re)prove the EF-emptiness problem for  
 13 bounded PTAs and then further related results, and found in [4]. We describe it in details,  
 14 as we will modify it in the subsequent proofs.

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

21 We initialize the clocks with the gadget in Figure 2a (that also blocks the case where  
 22  $a = 0$ ). Clearly, when in  $q_0$  with  $x = 0$ , we have  $y = z = 1$ , which indeed corresponds to  
 23 counter values 0.

24 We now present the gadget encoding the increment instruction of  $C_1$  in Figure 2b. The  
 25 transition from  $q_i$  to  $l_{i1}$  only serves to clearly indicate the entry in the increment gadget and  
 26 is done in 0 time unit. Since we use only equalities, there are really only two paths that go  
 27 through the gadget: one going through  $l_{i2}$  and one through  $l'_{i2}$ . Let us begin with the former.  
 28 We start from some encoding configuration:  $x = 0$ ,  $y = 1 - ac_1$  and  $z = 1 - ac_2$  in  $q_i$  (and  
 29 therefore the same in  $l_{i1}$ ). We can enter  $l_{i2}$  (after elapsing enough time) if  $1 - ac_2 \leq 1$ , *i. e.*,  
 30  $ac_2 \geq 0$ , which implies that  $a \geq 0$ , and when entering  $l_{i2}$  we have  $x = ac_2$ ,  $y = 1 - ac_1 + ac_2$   
 31 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  
 32 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  
 33 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  $z = 1 - ac_2$ ,  
 34 as expected.

35 We now examine the second path. We can enter  $l'_{i2}$  if  $1 - ac_1 \leq a + 1$ , *i. e.*,  $a(c_1 + 1) \geq 0$ ,  
 36 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

1 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  
 2 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  
 3 entering  $q_j$  we have  $x = 0$ ,  $y = 1 - a(c_1 + 1)$  and  $z = 1 - ac_2$ , as expected.

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

7 Decrement is done similarly by replacing guards  $y = a + 1$  with  $y = 1$ , and guards  $x = 1$   
 8 and  $z = 1$  with  $x = a + 1$  and  $z = a + 1$ , respectively.

9 From  $q_i$ , to encode zero-testing  $C_1$  and going to  $q_j$ , we only need to add a transition  
 10 from  $q_i$  to  $q_j$  with guard  $y = 1 \wedge x = 0$ .

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

12 The action associated with the transitions do not matter; we can assume a single action  $\sigma$   
 13 on all transitions (omitted in all figures).

14 Finally, we add a self-loop (with no guard) on the location  $q_{\text{halt}}$ , ensuring that whenever  
 15  $q_{\text{halt}}$  is reachable then there exists an infinite run in the PTA.

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

19 **1.** either the value of the counters is not bounded. Then, for any parameter valuation, at  
 20 some point during an incrementation of, say,  $C_1$  we will have  $a(c_1 + 1) > 1$  when taking  
 21 the transition from  $l_{i2}$  to  $l_{i3}$  and the PTA will be blocked. Therefore, there exists no  
 22 parameter valuation for which there exists an infinite run.

23 **2.** or the value of the counters remains bounded. Let  $c$  be their maximal value. Let us  
 24 consider two subcases:

25 **a.** either the machine reaches  $q_{\text{halt}}$ : in that case, if  $c = 0$  and  $0 < a \leq 1$  or  $c > 0$  and  
 26  $ca < 1$ , then the PTA valuated with such parameter valuations correctly simulates  
 27 the machine, yielding a (unique) run reaching location  $q_{\text{halt}}$ . From there, this run is  
 28 infinite thanks to the self-loop on  $q_{\text{halt}}$ . The set of such valuations for  $a$  is certainly  
 29 non-empty:  $a = \frac{1}{2}$  belongs to it if  $c = 0$  and  $a = \frac{1}{c}$  does otherwise.

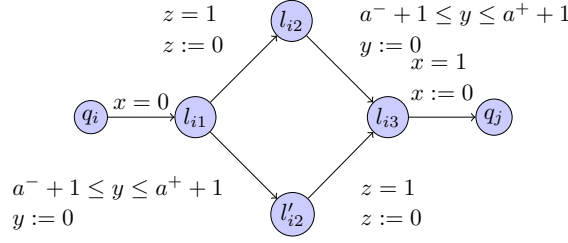
30 **b.** or the machine does not halt. Then again, for a sufficiently small parameter valuation  
 31 (*i. e.*,  $a < 1$  if  $c = 0$  and  $a \leq \frac{1}{c}$  otherwise), the machine is properly simulated, and since  
 32 the machine does not halt, then the run simulating the infinite execution is infinite  
 33 too. For other values of  $a$ , the machine will block at some point in an increment  
 34 gadget, because  $a$  is not small enough and the guard to  $q_j$  cannot be satisfied.

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

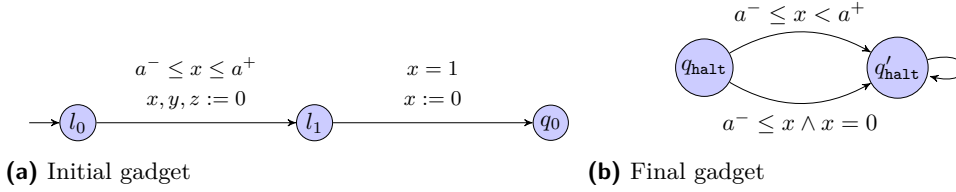
36 Hence the value of the counters remains bounded iff there exists a parameter valuation  $v$   
 37 such that  $v(\mathcal{A})$  contains an infinite run. ◀

38 ▶ **Remark.** Throughout this paper, we allow guards and invariants of the form  $x \sim$   
 39  $\sum_{1 \leq j \leq M} \beta_j p_j + d$ , which is more restrictive than [6] (that allows parametric coefficients  
 40 different from 0 and 1, as well as diagonal constraints), but more permissive than [2], that  
 41 only allows a syntax  $x \sim p$ . In fact, most papers in the literature define their own syntax  
 42 (see [3] for a survey). We can adapt our proof to fit in the most restrictive syntax ( $x \sim p$ )  
 43 as follows: transitions with  $y = a + 1$  guards and  $y := 0$  reset can be equivalently replaced  
 44 by one transition with a  $y = 1$  guard and a reset of some additional clock  $w$ , followed by  
 45 a transition with a  $w = a$  guard and the  $y := 0$  reset (and similarly for  $x$  and  $z$  is the





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



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

1 decrement gadget). This also allows the proof to work without complex parametric expres-  
 2 sions in guards, using three additional clocks (we conjecture that a smarter encoding can be  
 3 exhibited to factor these additional clocks, so as to use a single additional clock). A similar  
 4 modification can be applied to all subsequent undecidability proofs.

5 **4 Deadlock-Existence-Emptiness**

6 ► **Theorem 10.** *The deadlock-existence-emptiness problem is undecidable for closed bounded*  
 7 *L/U-PTAs, with 3 clocks and 2 parameters.*

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

9 Let us consider the encoding used in the proof of Theorem 9, that we transform into an  
 10 L/U-PTA by replacing any comparison of a clock with  $a$  (say  $x = a$ ) into  $x \leq a^+ \wedge x \geq a^-$ ,  
 11 where  $a^-$  (resp.  $a^+$ ) is a lower-bound (resp. upper-bound) parameter. We give the modified  
 12 increment gadget in Figure 3 (other gadgets are modified in a similar fashion).

13 We replace the initial gadget (Figure 2a) with the new one in Figure 4a. Before initializing  
 14 the values of the counters, this gadget first ensures that  $a^- \leq a^+$ .

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

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

22 Let us show that there exists a parameter valuation for which the system contains at  
 23 least one deadlock iff the 2-counter machine halts, which is undecidable [13]. Let us reason  
 24 by cases on the valuations of  $a^-$  and  $a^+$ .

- 25 **1.** If  $a^- > a^+$ , the initial gadget cannot be passed, but thanks to the unguarded transitions  
 26 to  $q'_{\text{halt}}$ , all runs eventually end in  $q'_{\text{halt}}$ , from which the absence of deadlock is guaranteed  
 27 by the unguarded self-loop.

- 1 2. If  $a^- < a^+$ , the machine may not be properly simulated because some transitions do not  
 2 occur at the right time and some run could reach  $q_{\text{halt}}$  while the machine does not halt.  
 3 Let us consider a run in the TA obtained with such a parameter valuation.
- 4 a. either this run is infinite and remains in the machine (*e. g.*, it loops infinitely through  
 5 the increment, decrement and 0-test gadgets of our encoding). Then there is no  
 6 deadlock.
- 7 b. or this run would block in a gadget (were it not for the unguarded transitions to  
 8  $q'_{\text{halt}}$ , due to a guard that cannot be satisfied); in that case, thanks to the unguarded  
 9 transitions to  $q'_{\text{halt}}$ , this run can go to  $q'_{\text{halt}}$ , from which it is deadlock-free.
- 10 c. or this run reaches  $q_{\text{halt}}$ ; from there, thanks to the upper transition in Figure 4b, it  
 11 can reach  $q'_{\text{halt}}$ , from which it is again deadlock-free.
- 12 3. If  $a^- = a^+ = 0$ , the machine may again not be properly simulated: again we could  
 13 reach  $q_{\text{halt}}$  while the machine does not halt. The situation is similar to the previous case  
 14 ( $a^- < a^+$ ) except that in  $q_{\text{halt}}$  a run has to take the lower transition in Figure 4b to  
 15 reach  $q'_{\text{halt}}$ , from which it is again deadlock-free.
- 16 4. If  $a^- = a^+ > 0$ :
- 17 a. Either the machine does not halt:
- 18 i. ... and the counters remain bounded: for some parameter valuations small enough  
 19 to encode the value of the counters (typically  $a^- = a^+ \leq \frac{1}{c}$ , where  $c$  is the maximum  
 20 value of both  $C_1$  and  $C_2$ ) then the PTA correctly simulates the infinite execution  
 21 of the machine, and the system is deadlock-free. (Note that such valuations can  
 22 also lead to  $q'_{\text{halt}}$  anytime, but this is harmless since this location guarantees the  
 23 absence of deadlocks.) For other valuations, at some point we have  $a^-c_1 > 1$ ; more  
 24 specifically, there is an incrementation of  $C_1$  such that  $a^-c_1 \leq 1$  and  $a^-(c_1+1) > 1$ .  
 25 Hence, the run cannot continue in the encoding, but can reach  $q'_{\text{halt}}$ , from where  
 26 the run is non-blocking.
- 27 ii. ... and the counters are unbounded. Then whatever the value of  $a^- > 0$ , at  
 28 some point we have  $a^-c_1 > 1$ . Then, when executing the corresponding increment  
 29 gadget,  $q'_{\text{halt}}$  can be reached from  $l_{i2}$ , from where the run is non-blocking.
- 30 Hence if the machine does not halt, the system is deadlock-free for all parameter  
 31 valuations.
- 32 b. Or the machine halts. In this case, if  $c$  is the maximum value of both  $C_1$  and  $C_2$   
 33 over the (necessarily finite) halting execution of the machine, and if  $c > 0$ , then  
 34 for valuations such that  $a^- = a^+ \leq \frac{1}{c}$ , then there exists one run that correctly  
 35 simulates the machine (beside plenty of runs that will go to  $q'_{\text{halt}}$  due to the unguarded  
 36 transitions); this run that correctly simulates the machine eventually reaches  $q_{\text{halt}}$ .  
 37 From  $q_{\text{halt}}$ , for such valuations, the system is deadlocked: indeed, the transitions  
 38 from  $q_{\text{halt}}$  to  $q'_{\text{halt}}$  can only be taken if  $a^- < a^+$  or  $a^- = 0$ . The set of such valuations  
 39 for which there exists a run that correctly simulates the machine is certainly non-  
 40 empty:  $a^- = a^+ = \frac{1}{c}$  belongs to it (if  $c = 0$  then we choose, *e. g.*,  $a^- = a^+ = \frac{1}{2}$ ).  
 41 Hence, if the 2-counter machine halts, there exist parameter valuations for which a  
 42 run has no discrete successor, and hence the system is not deadlock-free.

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

45 ▶ **Corollary 11.** *The deadlock-existence-emptiness problem is undecidable for open bounded*  
 46 *L/U-PTAs, L/U-PTAs, bounded PTAs and PTAs, with 3 clocks and 2 parameters.*

1 **Proof.** Let us consider each formalism:

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

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

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

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

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

## 12 5 EG-Emptiness

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

15 ▶ **Theorem 12.** *The EG-emptiness problem is decidable for closed bounded L/U-PTAs.*

### 16 Symbolic Semantics

17 Let us first recall the symbolic semantics of PTAs (see, *e.g.*, [10]).

18 We define the *time elapsing* of a constraint  $C$ , denoted by  $C^\nearrow$ , as the constraint over  $X$   
 19 and  $P$  obtained from  $C$  by delaying all clocks by an arbitrary amount of time. We define  
 20 the *past* of  $C$ , denoted by  $C^\swarrow$ , as the constraint over  $X$  and  $P$  obtained from  $C$  by letting  
 21 time pass backward by an arbitrary amount of time (see [10]). Given  $R \subseteq X$ , we define the  
 22 *reset* of  $C$ , denoted by  $[C]_R$ , as the constraint obtained from  $C$  by resetting the clocks in  $R$ ,  
 23 and keeping the other clocks unchanged. We denote by  $C \downarrow_P$  the projection of  $C$  onto  $P$ ,  
 24 *i.e.*, obtained by eliminating the clock variables (*e.g.*, using Fourier-Motzkin).

25 A *parametric zone* is a convex polyhedron over  $X \cup P$  in which all constraints on variables  
 26 are of the form  $x \sim plt$ , (parametric rectangular constraints) or  $x_i - x_j \sim plt$  (parametric  
 27 diagonal constraints), where  $x_i \in X$ ,  $x_j \in X$  and  $plt$  is a parametric linear term over  $P$ ,  
 28 *i.e.*, a linear term without clocks ( $\alpha_i = 0$  for all  $i$ ).

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

31 The symbolic semantics relies on the **Succ** operation. Given a symbolic state  $\mathbf{s} = (l, C)$   
 32 and an edge  $e = (l, g, \sigma, R, l')$ ,  $\text{Succ}(\mathbf{s}, e) = (l', C')$ , with  $C' = (([C \wedge g])_R \wedge I(l'))^\nearrow \wedge I(l')$ .  
 33 The **Succ** operation is effectively computable, using polyhedra operations: note that the  
 34 successor of a parametric zone  $C$  is a parametric zone (see *e.g.*, [10]).

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

39 We can now come back to the proof of Theorem 12.

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

42 Lemma 4 ensures that the TA  $v_{\text{inf/sup}}(\mathcal{A})$ , where  $v_{\text{inf/sup}}$  is obtained by valuating lower-  
 43 bound parameters  $p^-$  by  $m^-(p)$  and upper-bound parameters  $p^+$  by  $m^+(p)$ , includes all the  
 44 runs that could be produced with other parameter valuations. In  $v_{\text{inf/sup}}(\mathcal{A})$ , it is decidable

1 to find an infinite path staying in  $G$ , or conclude that none exist [1]. If we do find such a  
 2 path, we can terminate by answering yes to the EG-emptiness problem.

3 If we do not, then, in  $v_{\text{inf/sup}}(\mathcal{A})$ , all paths staying in  $G$  are finite. If we keep only discrete  
 4 actions and locations, which are in finite number, those paths therefore form a finite tree.  
 5 Let us recall again that, thanks to Lemma 4, all the discrete paths that stay in  $G$  and can  
 6 be obtained with any parameter valuation, belong to that tree.

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

15 If we find a deadlock, then we can terminate and answer yes to the EG-emptiness prob-  
 16 lem. Otherwise, we can terminate and answer no, because we have checked all the potential  
 17 discrete paths staying in  $G$  for any parameter valuation. ◀

18 Note that this proof fails when the L/U-PTA is not bounded or closed: consider the  
 19 L/U-PTA in Figure 1b. As  $p$  grows, there are more and more discrete behaviors, but there  
 20 is no cycle for any parameter valuation. In [6], the authors provide a finite upper bound  $N_{\mathcal{A}}$   
 21 for the upper-bound parameters such that if there exists a valuation such that the valuated  
 22 L/U-PTA has an accepting run, then the valuation giving 0 to lower bound parameters and  
 23  $N_{\mathcal{A}}$  to upper-bound parameters also ensures the existence of an accepting run. That bound  
 24 used in this example would indeed prove the non-existence of a cycle for any parameter  
 25 value, but it does not in turn allow us to derive a finite tree containing all the discrete  
 26 behaviors, for any possible parameter value (a larger bound would still give more runs).

27 Similarly, now consider the L/U-PTA in Figure 1c. If 0 is excluded from the domain  
 28 of  $p$ , we have a behavior similar to the previous example: as  $p$  gets closer and closer to 0, we  
 29 have more and more discrete behaviors. And even if we could derive a lower bound à la [6]  
 30 ensuring the non-existence of a cycle here, it would not give a finite tree of all the possible  
 31 discrete behaviors, for any parameter value.

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

35 ▶ **Theorem 13.** *The EG-emptiness problem is undecidable for open bounded L/U-PTAs,*  
 36 *with 4 clocks and 4 parameters.*

37 **Proof (sketch).** The proof reduces from the halting problem of a 2-counter machine. We  
 38 reuse the encoding of Theorem 10, and modify it so that the 2-counter machine executes in  
 39 a constant 1-time unit duration. This is achieved by replacing any occurrence of “1” in the  
 40 encoding with a (new) parameter, either  $b^-$  or  $b^+$  (depending on whether the occurrence  
 41 of 1 occurs as a lower-bound or an upper-bound); hence the duration of an increment or  
 42 decrement gadget is now at least  $b^-$  and at most  $b^+$  (by modifying a bit the construction  
 43 for the zero-test). Then, we add a global invariant  $w \leq 1$  (where  $w$  is a new clock) to all  
 44 locations. Now that the duration of any gadget is at least  $b^-$ , we therefore have that, for  
 45 any valuation  $b^- > 0$ , the number of operations the machine can perform is finite due to  
 46 the global invariant  $w \leq 1$ . We assume  $a^-, a^+, b^+ \in [0, 1]$  and  $b^- \in (0, 1]$ .

1 We prove that the 2-counter machine halts iff the set of valuations satisfying  $EG(L \setminus$   
 2  $\{q'_{\text{halt}}\})$  is not empty. We rule out valuations such that  $a^- > a^+$  or  $b^- > b^+$  by sending  
 3 them directly to  $q'_{\text{halt}}$ . For valuations  $a^- < a^+$  or  $b^- < b^+$ , the machine may not be  
 4 correctly simulated: either the encoding loops, and then blocks after some operations (due  
 5 to the invariant) which leads to  $q'_{\text{halt}}$ ; or it reaches  $q_{\text{halt}}$ , and goes to  $q'_{\text{halt}}$  thanks to an  
 6 appropriate gadget. Finally, valuations  $a^- = a^+$  and  $b^- = b^+ > 0$  may simulate correctly  
 7 the machine: if these valuations are not small enough, an increment will block, leading to  
 8  $q'_{\text{halt}}$ ; otherwise, for some valuations sufficiently small, and only if the machine halts, then  
 9  $q_{\text{halt}}$  is reached, and from there no transition leads to  $q'_{\text{halt}}$ , ensuring  $EG(L \setminus \{q'_{\text{halt}}\})$ .

10 See Appendix C for a detailed proof. ◀

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

13 We now prove that EG-emptiness is also undecidable for unbounded L/U-PTAs. When  
 14 not considering L/U-PTAs, proving an undecidability result for bounded PTAs usually gives  
 15 the undecidability for unbounded PTAs, as a bounded PTA can be simulated using a PTA  
 16 (by, *e. g.*, adding the bounds as a guard between a fresh location prior to the initial location  
 17 and the initial location, *e. g.*,  $p \in [\text{inf}, \text{sup}]$  becomes  $\text{inf} \leq x \leq \text{sup} \wedge p = x$ ). This may not be  
 18 true for L/U-PTAs, as such a construction requires to compare the clock and the parameter  
 19 using an equality. In addition, our proof for unbounded L/U-PTAs uses one parameter less  
 20 than for open bounded L/U-PTAs.

21 ▶ **Theorem 14.** *The EG-emptiness problem is undecidable for L/U-PTAs with 4 clocks and*  
 22 *3 parameters.*

23 **Proof (sketch).** We again use a reduction from the halting problem of a 2-counter machine.  
 24 Our proof essentially relies on a mechanism similar to the proof of Theorem 13; however, we  
 25 must use a different PTA encoding (the encoding used in the proof of Theorem 13 does not  
 26 work for unbounded L/U-PTAs, as it strongly relies on the fact that  $b^-$  be strictly positive).  
 27 Instead, we propose an encoding inspired by that of a 2-counter machine proposed in [5] to  
 28 prove the undecidability of the EF-emptiness problem for PTAs with a single integer-valued  
 29 parameter (that can also be rational-valued). We modify the encoding of [5] to obtain an  
 30 L/U-PTA, by splitting the single parameter  $a$  into a lower-bound parameter  $a^-$  and an  
 31 upper-bound parameter  $a^+$ , in the spirit of previous undecidability results for L/U-PTAs  
 32 in this paper (Theorems 10 and 13). Then, we add a global invariant  $w \leq b^+$  (where  $w$  is  
 33 a fresh clock never reset, and  $b^+$  a fresh upper-bound parameter), to ensure that, for any  
 34 valuation of  $b^+ > 0$ , the number of operations the machine can perform is finite (which  
 35 requires some modifications of the gadgets to ensure that they require at least 1 time unit).  
 36 The proof then follows a reasoning similar to that of Theorem 13.

37 See Appendix D for a detailed proof. ◀

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

## 41 6 Conclusion

42 Despite the vast number of undecidability results linked to the formalism of parametric  
 43 timed automata, and to which we also contribute here, we have achieved some decidability  
 44 for the existential parametric problem on the EG liveness property. This could be done

1 by imposing original constraints to the classical subclass of L/U-PTAs, pertaining to the  
 2 topology of the domain of the parameter values. This domain should be a closed and  
 3 bounded hyperrectangle of the rational space.

4 The subclass together with the EG property really lies on the boundary of decidability:  
 5 on the one hand, we have proved that considering unbounded, or bounded but open domains  
 6 leads again to undecidability for EG. On the other hand, if we consider — instead of the  
 7 EG property which asks for the existence of a maximal finite or infinite path staying in  
 8 some locations — only infinite maximal paths (existence of discrete cycles), then we have  
 9 proved that the problem becomes consistently decidable (with bounded domains or not).  
 10 And finally, if we consider only finite maximal paths (existence of deadlocks), then we have  
 11 proved that the problem becomes consistently undecidable.

12 Future work includes extending the EG decidability result to shapes other than hy-  
 13 perrectangles and studying actual synthesis. In addition, the decidability of problems we  
 14 proved undecidable for L/U-PTAs should be studied for two subclasses of L/U-PTAs, where  
 15 all parameters are upper bounds (U-PTAs) or all lower bounds (L-PTAs).

16 **Acknowledgements.** The authors thank Olivier H. Roux for fruitful discussions on the  
 17 topic of parametric timed automata.

## 18 ——— References ———

- 19 **1** Rajeev Alur and David L. Dill. A theory of timed automata. *Theoretical Computer Science*,  
 20 126(2):183–235, 1994.
- 21 **2** Rajeev Alur, Thomas A. Henzinger, and Moshe Y. Vardi. Parametric real-time reasoning.  
 22 In *STOC*, pages 592–601. ACM, 1993.
- 23 **3** Étienne André. What’s decidable about parametric timed automata? In *FTSCS*, volume  
 24 596 of *Communications in Computer and Information Science*, pages 1–17. Springer, 2015.
- 25 **4** Étienne André, Didier Lime, and Olivier H. Roux. Decision problems for parametric timed  
 26 automata. Technical report, 2016. URL: [www.lipn13.fr/t/PTAs.pdf](http://www.lipn13.fr/t/PTAs.pdf).
- 27 **5** Nikola Beneš, Peter Bezděk, Kim G. Larsen, and Jiří Srba. Language emptiness of  
 28 continuous-time parametric timed automata. In *ICALP, Part II*, volume 9135 of *LNCS*,  
 29 pages 69–81. Springer, 2015.
- 30 **6** Laura Bozzelli and Salvatore La Torre. Decision problems for lower/upper bound paramet-  
 31 ric timed automata. *Formal Methods in System Design*, 35(2):121–151, 2009.
- 32 **7** Edmund M. Clarke, E. Allen Emerson, and A. Prasad Sistla. Automatic verification of  
 33 finite-state concurrent systems using temporal logic specifications. *ACM Transactions on*  
 34 *Programming Languages and Systems*, 8(2):244–263, 1986.
- 35 **8** Laurent Doyen. Robust parametric reachability for timed automata. *Information Process-*  
 36 *ing Letters*, 102(5):208–213, 2007.
- 37 **9** Thomas Hune, Judi Romijn, Mariëlle Stoelinga, and Frits W. Vaandrager. Linear para-  
 38 metric model checking of timed automata. *JLAP*, 52-53:183–220, 2002.
- 39 **10** Aleksandra Jovanović, Didier Lime, and Olivier H. Roux. Integer parameter synthesis for  
 40 timed automata. *IEEE TSE*, 41(5):445–461, 2015.
- 41 **11** Leslie Lamport. Proving the correctness of multiprocess programs. *IEEE TSE*, 3(2):125–  
 42 143, 1977.
- 43 **12** Joseph S. Miller. Decidability and complexity results for timed automata and semi-linear  
 44 hybrid automata. In *HSCC*, volume 1790 of *LNCS*, pages 296–309. Springer, 2000.
- 45 **13** Marvin L. Minsky. *Computation: finite and infinite machines*. Prentice-Hall, Inc., 1967.

1 **A Proof of Theorem 6**

2 **Theorem 6 (recalled).** *The cycle-existence-emptiness problem is decidable for closed bounded L/U-PTAs.*

3 **Proof.** Recall that, thanks to the monotonicity property of L/U-PTAs (recalled in Lemma 4),  
4 any run possible for a valuation  $v$  of the parameters is also possible for any valuation of the  
5 parameters for which the upper-bound (resp. lower-bound) parameters are larger (resp.  
6 smaller) than or equal to that of  $v$ .

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

- 10 1. If  $v_{inf/sup}(\mathcal{A})$  contains an infinite run (which can be checked in PSPACE [1]), then since  
11  $\mathcal{A}|_{bounds}$  is closed,  $v_{inf/sup}$  belongs to  $bounds$ , and hence the set of parameter valuations  
12 that yield an infinite run is not empty.  
13 2. On the contrary, if  $v_{inf/sup}(\mathcal{A})$  contains no infinite run, then from the monotonicity  
14 property of L/U-PTAs (Lemma 4), no other valuation in  $bounds$  gives a TA with an  
15 infinite run, as such a TA could only contain less runs. Hence the set of parameter  
16 valuations that yield an infinite run is empty.

17

18 **B Proof of Lemma 7**

19 **Theorem 7 (recalled).** *Given an L/U-PTA  $\mathcal{A}$  and a subset of its locations  $G$ , the problem of the existence of at least one parameter valuation  $v$  such that  $v(\mathcal{A})$  has a run passing infinitely often through  $G$  is PSPACE-complete.*

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

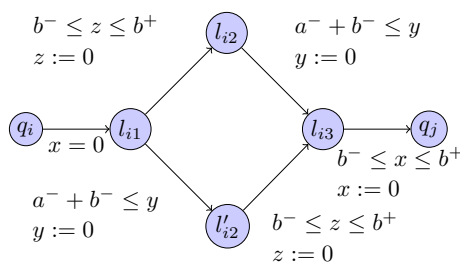
22  $\Leftarrow$  Considering an integer valuation is also a rational-valued valuation, the result trivially  
23 holds.

24  $\Rightarrow$  Assume there exists a rational-valued parameter valuation  $v$  for which  $v(\mathcal{A})$  contains  
an infinite run passing infinitely often through locations of  $G$ . Let  $v'$  be the integer  
parameter valuation obtained from  $v$  as follows:

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

24 From the monotonicity property of L/U-PTAs (Lemma 4), if  $v(\mathcal{A})$  yields an infinite run  
25 passing infinitely often through locations of  $G$ , then  $v'(\mathcal{A})$  does too.

26 Now, in [6, Theorem 8], it is proved that the problem of the the emptiness of the set of  
27 integer parameter valuations for which there exists an infinite run passing infinitely often  
28 through  $G$  is PSPACE-complete. This concludes the proof.  $\blacktriangleleft$



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

## C Proof of Theorem 13

**Theorem 13 (recalled).** *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 10, to which we will perform several modifications.

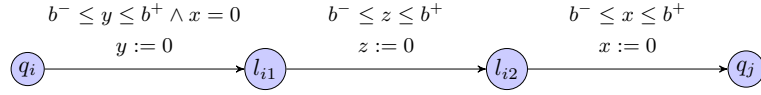
First, we force the 2-counter machine to execute in a constant 1-time unit duration as follows:

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.
2. We modify the zero-test 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  $w \leq 1$ , where  $w$  is a fresh clock that is never reset in the increment/decrement/zero-test gadgets. (These invariants are omitted in Figure 5.)

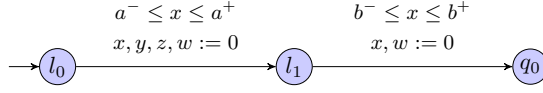
Hence, the duration of any gadget is at least  $b^-$  and therefore for any valuation  $b^- > 0$  the number of operations the machine can perform is finite due to the global invariant  $w \leq 1$ .

Then, before starting the 2-counter machine encoding, we add an initial gadget given in Figure 7. This gadget constrains  $a^- \leq a^+$ ,  $b^- \leq b^+$ , and is such that when leaving the





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



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

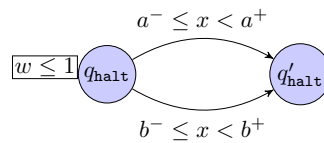
1 gadget then  $y, z \in [b^- \leq b^+]$  while  $x, w$  are 0. When  $b^- = b^+$ , this correctly encodes that  
 2 the value of both counters is 0.

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

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

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

- 14 1. If  $a^- > a^+$  or  $b^- > b^+$ , the initial gadget cannot be passed, and thanks to the transitions  
 15 to  $q'_{\text{halt}}$ , all runs eventually reach  $q'_{\text{halt}}$ , hence  $\text{EG}(L \setminus \{q'_{\text{halt}}\})$  does not hold.
- 16 2. If  $a^- < a^+$  and  $b^- \leq b^+$ , then the machine may not be correctly simulated: a given  
 17 run will either reach  $q_{\text{halt}}$ , in which case it will also reach  $q'_{\text{halt}}$  (as the guard from  
 18  $q_{\text{halt}}$  to  $q'_{\text{halt}}$  does not forbid this run), or it will loop in the machine until it eventually  
 19 gets blocked (since  $b^- > 0$  and because of the invariant  $w \leq 1$ , for any value of  $b^-$ ,  
 20 the maximal number of steps is  $\frac{1}{b^-}$ ); when being blocked, it has no other option than  
 21 going to  $q'_{\text{halt}}$ , thanks to the unguarded transitions from any location to  $q'_{\text{halt}}$ . Hence if  
 22  $a^- < a^+$ ,  $\text{EG}(L \setminus \{q'_{\text{halt}}\})$  does not hold.
- 23 3. If  $b^- < b^+$  (and  $a^- \leq a^+$ ), again the machine may not be correctly simulated, and  
 24 following a similar reasoning,  $\text{EG}(L \setminus \{q'_{\text{halt}}\})$  again does not hold.
- 25 4. If  $a^- = a^+$  and  $b^- = b^+ > 0$ :  
 26 a. Either the machine does not halt: in this case, after a maximum number of steps  
 27 (typically  $\frac{1}{b^-}$ ), a gadget will be blocked due to the invariant  $w \leq 1$ , and the run will  
 28 end in  $q'_{\text{halt}}$ . Hence if the 2-counter machine does not halt,  $\text{EG}(L \setminus \{q'_{\text{halt}}\})$  does not  
 29 hold.



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

1     **b.** Or the machine halts: in this case, if  $c$  is the maximum value of both  $C_1$  and  $C_2$  over  
2     the (necessarily finite) halting execution of the machine, and if  $m$  is the length of this  
3     execution, and if  $c > 0$ , then for valuations such that  $a^- = a^+ \leq \frac{b^-}{c}$  and  $b^- = b^+ \leq$   
4      $\frac{1}{m}$ , then there exists one run that correctly simulates the machine (beside plenty of  
5     runs that will go to  $q'_{\text{halt}}$  due to the unguarded transitions); this run that correctly  
6     simulates the machine eventually reaches  $q_{\text{halt}}$ . From  $q_{\text{halt}}$ , for such valuations, the  
7     system is deadlocked: indeed, the transitions from  $q_{\text{halt}}$  to  $q'_{\text{halt}}$  can only be taken  
8     if  $a^- < a^+$  or  $b^- < b^+$ . Hence  $\text{EG}(L \setminus \{q'_{\text{halt}}\})$  holds. The set of such valuations is  
9     certainly non-empty:  $a^- = a^+ = \frac{1}{m \times c}$  and  $b^- = b^+ = \frac{1}{m}$  belongs to it (if  $c = 0$  then  
10    we choose, *e. g.*,  $b^- = b^+ = 1$  and  $a^- = a^+ = \frac{1}{2}$ ). Hence, if the 2-counter machine  
11    halts, there exist parameter valuations for which  $\text{EG}(L \setminus \{q'_{\text{halt}}\})$  holds.

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

## 14 **D** Proof of Theorem 14

15 **Theorem 14 (recalled).** *The EG-emptiness problem is undecidable for L/U-PTAs with 4 clocks and 3 parameters.*

16 **Proof.** We will again use a reduction from the halting problem of a 2-counter machine. Our  
17 proof essentially relies on a mechanism similar to the proof of Theorem 13; however, we  
18 must use a different PTA encoding (the encoding used in the proof of Theorem 13 does not  
19 work for unbounded L/U-PTAs, as it strongly relies on the fact that  $b^-$  be strictly positive),  
20 which prevents us to factor the proof as much as we would have wished.

21 We propose here an encoding inspired by that of a 2-counter machine proposed in [5] to  
22 prove the undecidability of the EF-emptiness problem for PTAs with a single integer-valued  
23 parameter used to encode the maximum value of the two counters (although not considered  
24 in [5], the proof also works identically with a rational-valued parameter). The model of  
25 the 2-counter machine slightly differs from that considered in the rest of this paper. Two  
26 different instructions are considered:

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

29 Starting from the initial configuration  $(q_0, C_1 = 0, C_2 = 0)$  the machine either reaches  $q_{\text{halt}}$   
30 and halts, or loops forever. Knowing whether the machine halts is undecidable [13].

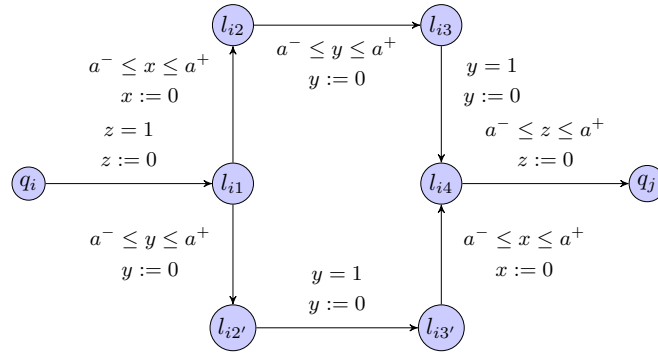
31 The encoding uses a single parameter  $a$ . Two clocks  $x$  and  $y$  are used to encode the value  
32 of the counters, while a third clock  $z$  is used as an auxiliary clock. Whenever  $z = 0$ , then  
33  $x = c_1$  and  $y = c_2$ .

34 We modify this encoding by splitting the single parameter  $a$  into a lower-bound param-  
35 eter  $a^-$  and an upper-bound parameter  $a^+$ , in the spirit of previous undecidability results  
36 for L/U-PTAs in this paper (Theorems 10 and 13).

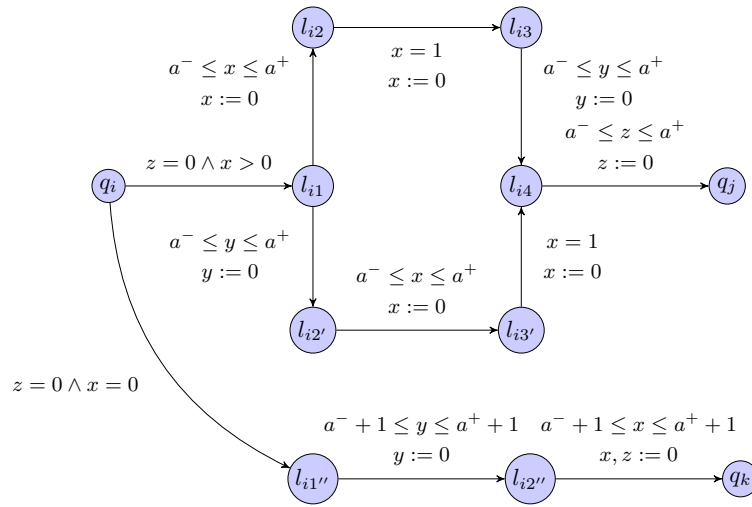
37 In addition, we request that the entire execution takes a time less than  $b^+$ , where  $b^+$   
38 is a fresh upper-bound parameter; this is achieved by adding an invariant  $w \leq b^+$  to all  
39 locations (with  $w$  a fresh clock never reset after the initial gadget).

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

43 The test and decrement gadget is similar, and given in Figure 10. We performed a  
44 slight modification to the zero-test of [5], that was executed in 0-time; we require in our



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

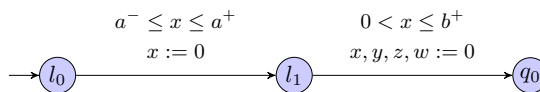


■ Figure 10 EG-emptiness for L/U-PTAs: test and decrement gadget

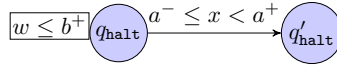
1 construction that each gadget takes at least one time unit. Hence, we rewrote it in Figure 10  
 2 so as to force at least one time unit to elapse after the clocks are tested, and so that the  
 3 final value of the clock is not changed, when  $a^- = a^+$  (in the spirit of the same operation  
 4 in the proof of Theorem 13): when performing the zero-test, we have  $x = z = 0$  and  $y = c_2$ .  
 5 Then after  $a - c_2 + 1$  time units (with  $a = a^+ = a^-$ ), we have  $x = z = a + 1 - c_2$  and  
 6  $y = a + 1$ , and we can take the transition to  $l_{i2''}$ , resetting  $y$ . Then after  $c_2$  time units, we  
 7 have  $x = z = a + 1$  and  $y = c_2$  and we can take the transition to  $l_{i2''}$ , resetting  $x$  and  $z$ . This  
 8 gives finally  $x = z = 0$  and  $y = c_2$  and the time spent in the gadget is in  $[a^- + 1, a^+ + 1]$ ,  
 9 and therefore is more than one time unit. Gadgets for the second counter are symmetric.

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

12 In addition, just as in Theorem 13, we add unguarded transitions from any location



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



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

1 (including that of the initial gadget, but excluding  $q_{\text{halt}}$ ) to a new location  $q'_{\text{halt}}$ . We also  
 2 add two transitions from  $q_{\text{halt}}$  to  $q'_{\text{halt}}$  given in the final gadget in Figure 12.

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

- 5 **1.** If  $a^- > a^+$  or  $b^+ = 0$ , the initial gadget cannot be passed: any run is sent to  $q'_{\text{halt}}$   
 6 because of the transitions to  $q'_{\text{halt}}$ , and therefore  $\text{EG}(L \setminus \{q'_{\text{halt}}\})$  does not hold.
- 7 **2.** If  $a^- < a^+$  and  $b^+ > 0$ , then the machine may not be correctly simulated: a given  
 8 run will either reach  $q_{\text{halt}}$ , in which case it will also reach  $q'_{\text{halt}}$  (as the guard from  $q_{\text{halt}}$   
 9 to  $q'_{\text{halt}}$  in Figure 12 does not forbid this run), or it will loop in the machine until it  
 10 eventually gets blocked: since  $b^+ > 0$ , since all gadgets require at least 1 time unit, for  
 11 any value of  $b^+$  the invariant  $z \leq b^+$  will eventually block a transition after at most  
 12  $b^+$  steps. When being blocked, a run has no other option than going to  $q'_{\text{halt}}$ , because  
 13 of the unguarded transitions from any location to  $q'_{\text{halt}}$ . Hence if  $a^- < a^+$  and  $b^+ > 0$ ,  
 14  $\text{EG}(L \setminus \{q'_{\text{halt}}\})$  does not hold.
- 15 **3.** Now, assume  $a^- = a^+$  and  $b^+ > 0$ .
  - 16 **a.** Either the machine does not halt: in this case, after a maximum number of steps  
 17 (typically at most  $b^+$ ), a gadget will be blocked due to the invariant  $z \leq b^+$ , and the  
 18 run will end in  $q'_{\text{halt}}$  because of the unguarded transitions from any location to  $q'_{\text{halt}}$ .  
 19 Hence if the 2-counter machine does not halt,  $\text{EG}(L \setminus \{q'_{\text{halt}}\})$  does not hold.
  - 20 **b.** Or the machine halts: in this case, if  $c$  is the maximum value of both  $C_1$  and  $C_2$  over  
 21 the (necessarily finite) halting execution of the machine, and if  $m$  is the length of this  
 22 execution, and if  $c > 0$ , then for valuations such that  $a^- = a^+ \leq c$  and sufficiently  
 23 large valuations of  $b^+$  (typically  $b^+ \geq m \times (a^+ + 1)$  as a gadget can take up to  $a^+ + 1$   
 24 time units), then there exists one run that correctly simulates the machine; this run  
 25 eventually reaches  $q_{\text{halt}}$ . From  $q_{\text{halt}}$ , for such values, the system is deadlocked. Hence,  
 26 if the 2-counter machine halts, there exist parameter valuations for which a run does  
 27 not reach  $q'_{\text{halt}}$ , *i. e.*, for which  $\text{EG}(L \setminus \{q'_{\text{halt}}\})$  holds.

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