

TCTL Model Checking Lower/Upper-Bound Parametric Timed Automata Without Invariants

Etienne André, Didier Lime, Mathias Ramparison

▶ **To cite this version:**

Etienne André, Didier Lime, Mathias Ramparison. TCTL Model Checking Lower/Upper-Bound Parametric Timed Automata Without Invariants. 16th International Conference on Formal Modeling and Analysis of Timed Systems (FORMATS 2018), Sep 2018, Beijing, China. pp.37-52, 10.1007/978-3-030-00151-3_3 . hal-01957138

HAL Id: hal-01957138
https://hal.science/hal-01957138

Submitted on 17 Dec 2018

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.

TCTL model checking lower/upper-bound parametric timed automata without invariants*

Étienne André[1], Didier Lime[2], Mathias Ramparison[1]

[1] Université Paris 13, LIPN, CNRS, UMR 7030, F-93430, Villetaneuse, France
[2] École Centrale de Nantes, LS2N, CNRS, UMR 6004, France

Abstract. We study timed systems in which some timing features are unknown parameters. First we consider Upper-bound Parametric Timed Automata (U-PTAs), one of the simplest extensions of timed automata with parameters, in which parameters are only used as clock upper bounds in the constraints. Up to now, there have been several decidability results for the existence of parameter values in U-PTAs such that flat TCTL formulas are satisfied. We prove here that this does not extend to the full logic and that only one level of nesting leads to undecidability. This provides, to the best of our knowledge, the first problem that is decidable for Timed Automata with an undecidable parametric emptiness version for U-PTAs. Second we study Lower/Upper-bound Parametric Timed Automata (L/U-PTAs) in which parameters are used either as clock lower bound, or as clock upper bound, but not both. We prove that without invariants, flat TCTL is decidable for L/U-PTAs by resolving the last non investigated liveness properties.

1 Introduction

Timed automata (TAs) [AD94] are a powerful formalism for modeling concurrent real-time systems; TAs extend finite-state automata with clocks, i. e., variables evolving at the same rate, that can be compared to integers in transition guards, and possibly reset to 0.

Despite notable successes in timed model checking, TAs become less suitable to model and verify systems when some timing constants are known with some imprecision—or completely unknown. Extending TAs with *timing parameters* (unknown constants) adds one more level of abstraction, and copes with uncertainty. When allowing parameters in place of integers in guards, TAs become parametric TAs (PTAs) [AHV93]. The model checking problem for TAs becomes a parametric model checking problem: given a PTA $\mathcal{A}$ and a formula φ (expressed in e. g., TCTL [ACD93]), what are the parameter valuations v such

* This is the author (and slightly extended) version of the manuscript of the same name published in the proceedings of the 16th International Conference on Formal Modeling and Analysis of Timed Systems (FORMATS 2018). The final version is available at dx.doi.org/10.1007/978-3-030-00151-3_3. This work is partially supported by the ANR national research program PACS (ANR-14-CE28-0002).

that the instance of $\mathcal{A}$ in which parameters are replaced using the values given by v (denoted $v(\mathcal{A})$) satisfies φ? In the PTA literature, the main problem studied is EF-emptiness ("is the set of valuations for which given location is reachable empty?"): it is "robustly" undecidable in the sense that, even when varying the setting, undecidability is preserved. For example, EF-emptiness is undecidable even for a single bounded parameter [Mil00], even for a single rational-valued or integer-valued parameter [BBLS15], even with only one clock compared to parameters [Mil00], or with strict constraints only [Doy07] (see [And17] for a survey). In contrast, decidability is ensured in some restrictive settings such as over discrete time with a single parametric clock (i. e., compared to parameters in at least one guard) [AHV93], or over discrete or dense time with one parametric clock and arbitrarily many non-parametric clocks [BO14,BBLS15], or over discrete time with two parametric clocks and a single parameter [BO14]. But the practical power of these restrictive settings remains unclear.

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]. L/U-PTAs enjoy mixed decidability results: while the EF-emptiness problem and the EF-universality problem ("Can we reach a given location, regardless of what valuations we give to the parameters?") are decidable, AF-emptiness ("is the set of valuations for which all runs eventually reach a given location empty?") is undecidable [JLR15]; as for EG-emptiness ("is the set of valuations for which one infinite or finite maximal run always remains in a given set of locations empty?"), it is decidable only when the parameter domain is bounded with closed bounds [AL17].

U-PTAs are L/U-PTAs with only upper-bound parameters [BL09], and are TAs' simplest parametric extension; since their introduction, no problem was ever shown undecidable for U-PTAs, when decidable for TAs, and all their known decidability results only came from the decidability for the larger class of L/U-PTAs. In [ALR16b], we showed that, in terms of union of untimed words, U-PTAs are not more expressive than TAs. A natural question is to investigate whether their expressiveness is anyhow beyond that of TAs, or whether the parametric emptiness version of all problems decidable for TAs remains decidable for U-PTAs.

Note that in [JLR13], the authors claim that AF-emptiness is undecidable for U-PTAs but the original unpublished proof had a fatal flaw, which is why the result was weakened to L/U-PTAs in [JLR15]. The result for U-PTAs is therefore still open.

Contribution Our first contribution is to show that the TCTL-emptiness problem ("given a TCTL formula, is the set of valuations v for which $v(\mathcal{A}) \models \varphi$ empty?") is undecidable for U-PTAs. This result comes in contrast with the fact that investigated flat TCTL formulas (namely EF, AG)—formulas that cannot be obtained by restraining another TCTL formula—are known to be decidable for U-PTAs, while others (EG and AF) are open. Our proof relies on the reduction

Class	U-PTAs without invariants	integer-valued L/U-PTAs without invariant	L/U-PTAs	bounded PTAs	PTAs
EF	[HRSV02]	[HRSV02]	[HRSV02]	*[Mil00]*	*[AHV93,Mil00]*
AF	open	**Theorem 3**	*[JLR15]*	*[ALR16a]*	*[JLR15]*
EG	open	**Theorem 3**	*[AL17]*	*[AL17]*	*[AL17]*
AG	[HRSV02]	[HRSV02]	[HRSV02]	*[ALR16a]*	*[ALR16a]*
flat TCTL	open	**Theorem 3**	*[JLR15]*	*[Mil00]*	*[AHV93]*
TCTL	***Theorem 1***	***Theorem 1***	*[JLR15]*	*[Mil00]*	*[AHV93]*

Table 1: Decidability of the emptiness problems for PTAs and subclasses

of the halting problem of a 2-counter machine to the emptiness of the $\mathsf{EGAF}_{=0}$ formula.

Our second contribution is that EG-emptiness is PSPACE-complete for (unbounded) integer-valued L/U-PTAs without invariants. Let us stress that EG-emptiness is undecidable for classical unbounded integer-valued L/U-PTAs with invariants [AL17], which draws a more accurate border between decidability and undecidability results regarding L/U-PTAs. Moreover, we show that EG-universality (also known as AF-emptiness) is PSPACE-complete for (unbounded) integer-valued L/U-PTAs without invariants, despite being undecidable for classical (rational- or integer-valued) L/U-PTAs with invariants [JLR15]. These results highlight the power invariants confer upon the expressiveness of L/U-PTAs. We deduce from all this that flat TCTL emptiness and universality is also decidable for integer-valued L/U-PTAs without invariants, which also makes the decidability frontier more precise with respect to nesting of TCTL formulas.

We give a summary of the known decidability results in Table 1, with our contributions in bold. We give from left to right the (un)decidability for U-PTAs, L/U-PTAs with integer-valued parameters without invariants, L/U-PTAs (the undecidability results also hold for integer-valued parameters), bounded PTAs (i. e., with a bounded parameter domain), and PTAs. We review the emptiness of TCTL subformulas (EF, AF, EG, AG), flat TCTL and full TCTL. Decidability is given in green, whereas undecidability is given in italic red. As U-PTAs can be seen as the simplest parametric extension of TAs, our undecidability result moves the undecidability frontier closer to TAs, and confirms that timed automata (while enjoying many decidability results) are a formalism very close to the undecidability frontier.

Outline Section 2 recalls the necessary preliminaries. Sections 3 and 4 show that TCTL-emptiness is undecidable for U-PTAs and bounded U-PTAs, respectively. Section 5 consists of the decidability results for integer-valued L/U-PTAs without invariants. Section 6 concludes the paper and proposes some perspectives.

2 Preliminaries

2.1 Clocks, parameters and guards

We assume a set $\mathbb{X} = \{x_1, \ldots, x_H\}$ of *clocks*, i. e., real-valued variables that evolve at the same rate. A clock valuation is a function $w : \mathbb{X} \to \mathbb{R}_+$. We identify a clock valuation w with the point $(w(x_1), \ldots, w(x_H))$ of $\mathbb{R}_+^H$. We write $\mathbf{0}$ for the clock valuation assigning 0 to all clocks. Given $d \in \mathbb{R}_+$, $w + d$ denotes the valuation s.t. $(w + d)(x) = w(x) + d$, for all $x \in \mathbb{X}$. Given $R \subseteq \mathbb{X}$, 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 $\mathbb{P} = \{p_1, \ldots, p_M\}$ of *parameters*, i. e., unknown constants. An upper-bound (resp. lower-bound) parameter p is such that, whenever it appears in a constraint $x \bowtie p + d$ with $d \in \mathbb{N}$ then necessarily $\bowtie \in \{\leq, <\}$ (resp. $\bowtie \in \{\geq, >\}$). A parameter *valuation* v is a function $v : \mathbb{P} \to \mathbb{Q}_+$. An *integer* parameter *valuation* v is a function $v : \mathbb{P} \to \mathbb{N}$. We identify a valuation v with the point $(v(p_1), \ldots, v(p_M))$ of $\mathbb{Q}_+^M$. A clock is *parametric* if it is compared at least once to a parameter, and *non-parametric* otherwise.

We assume $\bowtie \in \{<, \leq, =, \geq, >\}$, $\triangleleft \in \{<, \leq\}$. A *u*-guard g (resp. an *l*-guard g) is a conjunction of inequalities of the form $x \bowtie d$, or $x \triangleleft p + d$ with p an upper-bound parameter (resp. $p + d \triangleleft x$ with p a lower-bound parameter) and $d \in \mathbb{N}$. Given g, we write $w \models v(g)$ if the expression obtained by replacing each x with $w(x)$ and each p with $v(p)$ in g evaluates to true.

2.2 Lower/Upper-bound parametric timed automata

Let AP be a set of atomic propositions. Let us recall L/U-PTAs:

Definition 1 (L/U-PTA). *An L/U-PTA $\mathcal{A}$ is a tuple $\mathcal{A} = (\Sigma, L, \mathbf{L}, l_0, \mathbb{X}, \mathbb{P}, E)$, where:*

1. *Σ is a finite set of actions,*
2. *L is a finite set of locations,*
3. *$\mathbf{L}$ is a label function $\mathbf{L} : L \to 2^{AP}$,*
4. *$l_0 \in L$ is the initial location,*
5. *$\mathbb{X}$ is a finite set of clocks,*
6. *$\mathbb{P}$ is a finite set of parameters partitioned into lower-bound parameters and upper-bound parameters*
7. *E is a finite set of edges $e = (l, g, a, R, l')$ where $l, l' \in L$ are the source and target locations, $a \in \Sigma$, $R \subseteq \mathbb{X}$ is a set of clocks to be reset, and g is a conjunction of a* u*-guard and an* l*-guard.*

Unlike the classical definition of [HRSV02], we consider L/U-PTAs without invariants. We define a U-PTA [BL09] as an L/U-PTA where in each edge, g is a u-guard. Note that an L/U-PTA where we replace all guards are made of conjunctions of inequalities of the form $x \bowtie p$, or $x \bowtie d$, with $d \in \mathbb{N}$, becomes a PTA as defined in [AHV93].

Given a parameter valuation v, we denote by $v(\mathcal{A})$ the non-parametric structure where all occurrences of a parameter p_i have been replaced by $v(p_i)$. We denote as a *timed automaton* any structure $v(\mathcal{A})$, by assuming a rescaling of the constants: by multiplying all constants in $v(\mathcal{A})$ by their least common denominator, we obtain an equivalent (integer-valued) TA, as defined in [AD94]. A *bounded* U-PTA is a U-PTA with a bounded parameter domain that assigns to each parameter a minimum integer bound and a maximum integer bound. That is, each parameter p_i ranges in an interval $[a_i, b_i]$, with $a_i, b_i \in \mathbb{N}$. Hence, a bounded parameter domain is a hyperrectangle of dimension M.

Let us first recall the concrete semantics of TA.

Definition 2 (Semantics of a TA). *Given a L/U-PTA $\mathcal{A} = (\Sigma, L, \mathbf{L}, l_0, \mathbb{X}, \mathbb{P}, E)$, and a parameter valuation v, the semantics of $v(\mathcal{A})$ is given by the timed transition system (TTS) $(S, s_0, \rightarrow)$, with $S = \{(l, w) \in L \times \mathbb{R}_+^H\}$, $s_0 = (l_0, \mathbf{0})$ and $\rightarrow$ consists of the discrete and (continuous) delay transition relations:*

1. *discrete transitions: $(l, w) \overset{e}{\mapsto} (l', w')$, if $(l, w), (l', w') \in S$, and there exists $e = (l, g, a, R, l') \in E$, such that $w' = [w]_R$, and $w \models v(g)$.*
2. *delay transitions: $(l, w) \overset{d}{\mapsto} (l, w + d)$, with $d \in \mathbb{R}_+$.*

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

Given a TA $v(\mathcal{A})$ with concrete semantics $(S, s_0, \rightarrow)$, we refer to the states of S as the *concrete states* of $v(\mathcal{A})$. A *run* of $v(\mathcal{A})$ is a possibly infinite alternating sequence of states of $v(\mathcal{A})$ and edges starting from the initial state s_0 of the form $s_0 \xrightarrow{e_0} s_1 \xrightarrow{e_1} \cdots \xrightarrow{e_{m-1}} s_m \xrightarrow{e_m} \cdots$, such that for all $i = 0, 1, \ldots$, $e_i \in E$, and $(s_i, e_i, s_{i+1}) \in \rightarrow$. Given a run ρ, $\mathsf{time}(\rho)$ gives the total sum of the delays d along ρ. Given a state $s = (l, w)$, we say that s is reachable if s appears in a run of $v(\mathcal{A})$. By extension, we say that a label lb is reachable in $v(\mathcal{A})$ if there exists a state (l, w) that is reachable such that $lb \in \mathbf{L}(l)$. Given a set of locations $T \subseteq L$, 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 Timed CTL

TCTL [ACD93] is the quantitative extension of CTL where temporal modalities are augmented with constraints on duration. Formulae are interpreted over TTS.

Given $ap \in AP$ and $c \in \mathbb{N}$, a TCTL formula is given by the following grammar:

$$\varphi ::= \quad \top \quad | \quad ap \quad | \quad \neg\varphi \quad | \quad \varphi \wedge \varphi \quad | \quad \mathsf{E}\varphi\mathsf{U}_{\bowtie c}\varphi \quad | \quad \mathsf{A}\varphi\mathsf{U}_{\bowtie c}\varphi$$

A reads "always", E reads "exists", and U reads "until".

Standard abbreviations include Boolean operators as well as $\mathsf{EF}_{\bowtie c}\varphi$ for $\mathsf{E}\top\mathsf{U}_{\bowtie c}\varphi$, $\mathsf{AF}_{\bowtie c}\varphi$ for $\mathsf{A}\top\mathsf{U}_{\bowtie c}\varphi$ and $\mathsf{EG}_{\bowtie c}\varphi$ for $\neg\mathsf{AF}_{\bowtie c}\neg\varphi$. ($\mathsf{F}$ reads "eventually" while G reads "globally".)

Definition 3 (Semantics of TCTL). *Given a TA $v(\mathcal{A})$, the following clauses define when a state s_i of its TTS $(S, s_0, \to)$ satisfies a TCTL formula φ, denoted by $s_i \models \varphi$, by induction over the structure of φ (semantics of Boolean operators is omitted):*

1. *$s_i \models \mathsf{E}\varphi\mathsf{U}_{\bowtie c}\Psi$ if there is a maximal run ρ in $v(\mathcal{A})$ with $\sigma = s_i \xrightarrow{e_i} \cdots \xrightarrow{e_{j-1}} s_j$ $(i < j)$ a prefix of ρ s.t. $s_j \models \Psi$, $\mathsf{time}(\sigma) \bowtie c$, and if $i \leq k < j$, $s_k \models \varphi$, and*
2. *$s_i \models \mathsf{A}\varphi\mathsf{U}_{\bowtie c}\Psi$ if for each maximal run ρ in $v(\mathcal{A})$ there exists $\sigma = s_i \xrightarrow{e_i} \cdots \xrightarrow{e_{j-1}} s_j$ $(i < j)$ a prefix of ρ s.t. $s_j \models \Psi$, $\mathsf{time}(\sigma) \bowtie c$, and if $i \leq k < j$, $s_k \models \varphi$.*

In $\mathsf{E}\varphi\mathsf{U}_{\bowtie c}\Psi$ the classical until is extended by requiring that φ be satisfied within a duration (from the current state) verifying the constraint "$\bowtie c$". Given v, an L/U-PTA $\mathcal{A}$ and a TCTL formula φ, we write $v(\mathcal{A}) \models \varphi$ when $s_0 \models \varphi$.

We define *flat TCTL* as the subset of TCTL where, in $\mathsf{E}\varphi\mathsf{U}_{\bowtie c}\varphi$ and $\mathsf{A}\varphi\mathsf{U}_{\bowtie c}\varphi$, φ must be a formula of propositional logic (a boolean combination of atomic propositions).

In this article, we address the following problems:

TCTL-emptiness problem:
INPUT: an L/U-PTA $\mathcal{A}$ and a TCTL formula φ
PROBLEM: is the set of valuations v such that $v(\mathcal{A}) \models \varphi$ empty?

TCTL-universality problem:
INPUT: an L/U-PTA $\mathcal{A}$ and a TCTL formula φ
PROBLEM: are all valuations v such that $v(\mathcal{A}) \models \varphi$?

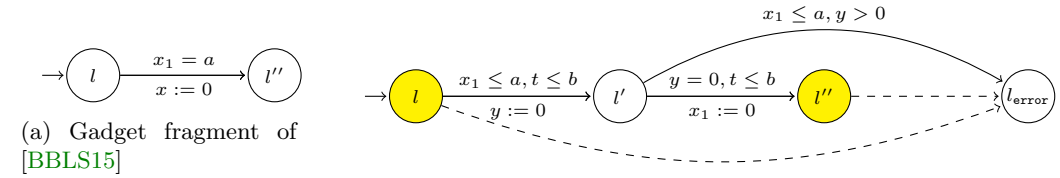
More specifically, we will address in Section 5 the EG-emptiness (resp. EG-universality problem) i. e., whether, given an L/U-PTA $\mathcal{A}$ and a subset of its locations T, the set of parameter valuations for which there is a run in $v(\mathcal{A})$ that stays in T is empty (resp. universal).

3 Undecidability of TCTL emptiness for U-PTAs

We exhibit here a formula that shows that TCTL emptiness is undecidable for U-PTAs.

Theorem 1. *The $\mathsf{EGAF}_{=0}$-emptiness problem is undecidable for U-PTAs.*

Proof. We reduce from the halting problem for two-counter machines, which is undecidable [Min67]. Recall that a two-counter machine is a finite state machine with two integer-valued counters $\mathsf{c}_1, \mathsf{c}_2$. Two different instructions (presented for c_1 and identical for c_2) are considered:



(a) Gadget fragment of [BBLS15]

(b) Modified gadget of [BBLS15] enforcing $\mathsf{EGAF}_{=0}\heartsuit$

Fig. 1: A gadget fragment and its modification into a U-PTA

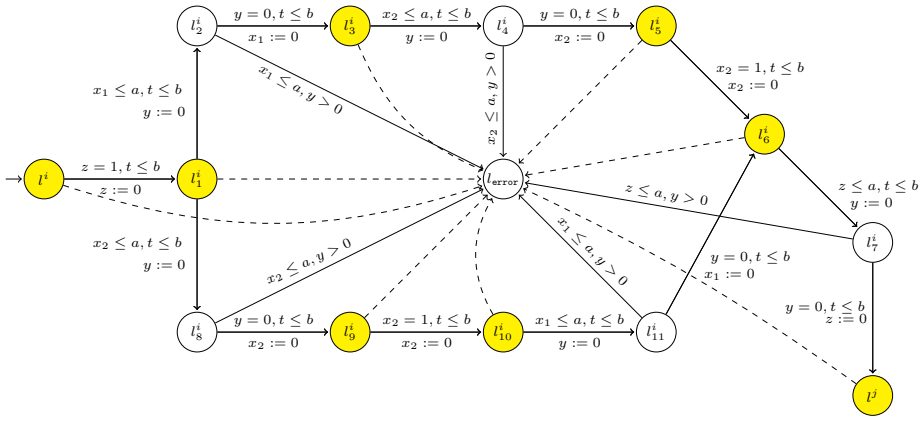


Fig. 2: increment gadget

1. when in state q_i, increment c_1 and go to q_j;
2. when in state q_i, if $\mathsf{c}_1 = 0$ go to q_k, otherwise decrement c_1 and go to q_j.

We assume w.l.o.g. that the machine halts iff it reaches a special state q_{halt}.

We define a U-PTA that, under some conditions, will encode the machine, and for which $\mathsf{EGAF}_{=0}\heartsuit$-emptiness holds iff the machine does not halt (for some $\heartsuit \in AP$). Our U-PTA $\mathcal{A}$ uses two (possibly integer-valued) parameters a, b, and five clocks i. e., a single non-parametric clock y and four parametric clocks x_1, x_2, z, t. We also omit the transition labels as they are not relevant for the emptiness problem. Each state q_i of the two-counter machine is encoded by a location l^i of $\mathcal{A}$. Each increment (resp. decrement) instruction of the two-counter machine is encoded into a U-PTA fragment depicted in Figs. 2 and 3, respectively.

Our encoding is inspired by [BBLS15] and is such that when in l^i with $w(z) = 0$ then $w(x_1)$ (resp. $w(x_2)$) represents the value of the counter c_1 (resp. c_2). However, as U-PTAs disallow constraints of the form $x = a$, we need to considerably modify the encoding. Each of our locations has exactly one label: $\heartsuit$ for the locations already present in [BBLS15] (depicted in yellow in our figures), and $\spadesuit$ for the newly introduced locations (depicted in white in our figures). In [BBLS15], the gadgets encoding the two-counter machine instruc-

tions use edges of the form of Fig. 1a. To define a proper U-PTA, we replace each of these edges by a special construction given in Fig. 1b using only inequalities of the form $x \leq a$. Our goal is to show that a run will exactly encode the two-counter machine if all guards $x \leq a$ are in fact taken when the clock valuation is exactly equal to a. Those runs are further denoted by $\rho_{\heartsuit}$. Consider the transformed version given in Fig. 1b: due to the $\leq$, runs exist that take the guard "too early" (i. e., before $x_1 = a$). Those are denoted by $\rho_{\spadesuit}$. But, in that case, observe that in l', one can either take the transition to l'' in 0-time, or spend some time in l' and then (with guard $y > 0$) go to $l_{\texttt{error}}$. Therefore on this gadget, $\mathsf{EGAF}_{=0}\heartsuit$ is true at l' iff the guard $x_1 \leq a$ from l to l' is taken at the very last moment. Note that $\mathsf{EGAF}_{=0}\heartsuit$ is trivially true in l and l'' as both locations are labeled with $\heartsuit$. (Also note that there are plenty of runs from l to $l_{\texttt{error}}$ that do not encode properly the machine; they will be discarded in our reasoning later.)

We also assume a condition $t \leq b$ on all guarded transitions, where t is a clock never reset. As presented in Fig. 1b, there are transitions without guard (dashed) from l, l'' (labeled with $\heartsuit$) to $l_{\texttt{error}}$. This is done to enforce the violation of $\mathsf{EGAF}_{=0}\heartsuit$ whenever $t = b$: indeed, while $t < b$ a run can either go to $l_{\texttt{error}}$ from a location labeled with $\heartsuit$, or not, but as $t = b$ every run is forced to go to $l_{\texttt{error}}$, making $\mathsf{EGAF}_{=0}\heartsuit$ false.

The gadgets presented in Figs. 2 and 3 provide an encoding to respectively increase and decrease the values of the counters of the two-counter machine.

Increment We give the increment gadget for c_1 in Fig. 2 (the gadget for c_2 is symmetric). Let v be a valuation, and assume we are in configuration (l^i, w), where $w(z) = 0$. First note that if $w(x_1) \geq v(a)$, there is no execution ending in l^j due to the delay of one time unit on the transition from l^i to l^i_1, and the guard $x_1 \leq a$ tested in both the upper and the lower branch in the automaton. The same reasoning is relevant for $w(x_2)$.

Assume $w(x_1), w(x_2) < v(a)$. Two cases show up: $w(x_1) \leq w(x_2)$ and $w(x_1) > w(x_2)$, which explains why we need two paths in Fig. 2. First, if $w(x_1) \leq w(x_2)$, we can perform several executions with different time delays, but those are bounded. In the following, we write w as the tuple $(w(x_1), w(x_2), w(z), w(y))$, omitting t.

From l^i, we prove that there is a unique run that reaches l^j without violating our property. It is the one that takes each transition with a u-guard $x \leq a$ at the exact moment $w(x) = v(a)$ which we describe in the following.

From (l^i, w), the unique delay to pass the transition is 1, hence we arrive in the configuration $(l^i_1, (w(x_1) + 1, w(x_2) + 1, w(y) + 1, 0))$. Here, the largest delay to pass the transition is $v(a) - w(x_1) - 1$ so a configuration we possibly obtain is $(l^i_2, (d_1, d_2, d_3, 0))$ with $(d_1, d_2, d_3) \leq (v(a), w(x_2) - w(x_1) + v(a), v(a) - w(x_1) - 1)$. If $(d_1, d_2, d_3) < (v(a), w(x_2) - w(x_1) + v(a), v(a) - w(x_1) - 1)$ then the guard $y > 0$ in the transition to $l_{\texttt{error}}$ is verified, hence our property $\mathsf{EGAF}_{=0}\heartsuit$ is violated. We remove all these runs and keep the only run that ends in the exact configuration $(l^i_2, (v(a), w(x_2) - w(x_1) + v(a), v(a) - w(x_1) - 1, 0))$. As $y = 0$ holds

the next configuration is $(l_3^i, (0, w(x_2) - w(x_1) + v(a), v(a) - w(x_1) - 1, 0))$. The largest delay to pass the next transition is $w(x_1) - w(x_2)$, so a configuration we possibly obtain is $(l_4^i, (d_1, d_2, d_3, 0))$ with $(d_1, d_2, d_3) \leq (w(x_1) - w(x_2), v(a), v(a) - w(x_2) - 1)$. If $(d_1, d_2, d_3) < (w(x_1) - w(x_2), v(a), v(a) - w(x_2) - 1)$ then the guard $y > 0$ in the transition to $l_{\texttt{error}}$ is verified, hence our property $\mathsf{EGAF}_{=0}\heartsuit$ is violated. We remove all these runs and keep the only run that ends in the exact configuration $(l_4^i, (w(x_1) - w(x_2), v(a), v(a) - w(x_2) - 1, 0)$. As $y = 0$ holds the next configuration is $(l_5^i, (w(x_1) - w(x_2), 0, v(a) - w(x_2) - 1, 0)$. Now the unique delay to pass the transition is 1, hence as we reset x_2 we arrive in the configuration $(l_6^i, (w(x_1) - w(x_2) + 1, 0, v(a) - w(x_2), 1)$. The largest delay to pass the next transition is $w(x_2)$, so a configuration we possibly obtain is $(l_7^i, (d_1, d_2, d_3, 0))$ with $(d_1, d_2, d_3) \leq (w(x_1) + 1, w(x_2), v(a))$. If $(d_1, d_2, d_3) < (w(x_1) + 1, w(x_2), v(a))$ then the guard $y > 0$ in the transition to $l_{\texttt{error}}$ is verified, hence our property $\mathsf{EGAF}_{=0}\heartsuit$ is violated. We remove all these runs and keep the only run that ends in the exact configuration $(l_7^i, (w(x_1) + 1, w(x_2), v(a), 0))$. As $y = 0$ holds the next configuration is $(l^j, (w(x_1) + 1, w(x_2), 0, 0))$, and as $w(z) = 0$, $w(x_1)$ represents the exact value of the counter c_1 increased by 1.

In its shorter form, this run is: $(l^i, w) \xrightarrow{1} (l_1^i, (w(x_1)+1, w(x_2)+1, w(y)+1, 0)) \xrightarrow{v(a)-w(x_1)-1} (l_2^i, (v(a), w(x_2) - w(x_1) + v(a), v(a) - w(x_1) - 1, 0)) \xrightarrow{0} (l_3^i, (0, w(x_2) - w(x_1) + v(a), v(a) - w(x_1) - 1, 0)) \xrightarrow{w(x_1)-w(x_2)} (l_4^i, (w(x_1) - w(x_2), v(a), v(a) - w(x_2) - 1, 0)) \xrightarrow{0} (l_5^i, (w(x_1) - w(x_2), 0, v(a) - w(x_2) - 1, 0)) \xrightarrow{1} (l_6^i, (w(x_1) - w(x_2) + 1, 0, v(a) - w(x_2), 1)) \xrightarrow{w(x_2)} (l_7^i, (w(x_1) + 1, w(x_2), v(a), 0)) \xrightarrow{0} (l^j, (w(x_1) + 1, w(x_2), 0, 0))$

Second, if $w(x_1) > w(x_2)$ we take the lower branch of the automaton and apply the same reasoning.

Decrement and 0-test The decrement and 0-test gadget is similar: we reuse the reasoning of [BBLS15], and apply the same modifications as in Fig. 1b. Note that the 0-test gadget has been completely rewritten from [BBLS15] to ensure a time elapsing of at least $a + 1$ time units when the guards are taken at the last moment.

We give the decrement gadget in Fig. 3. Assume we are in a configuration (l^i, w) where $w(z) = 0$ and suppose $w(x_1) > 0$. We can enter the configuration $(l_1, (w(x_1), w(x_2), 0, w(y)))$ as the guard $z = 0$ ensures no time has elapsed.

Two cases show up: $w(x_1) \leq w(x_2)$ and $w(x_1) > w(x_2)$.

First, if $w(x_1) \leq w(x_2)$, we can perform several executions with different time delays, but those are bounded. From l^i, there is a unique run that reaches l^j without violating our property. It is the one that takes each transition with a u-guard $x \leq a$ at the exact moment $w(x) = v(a)$:

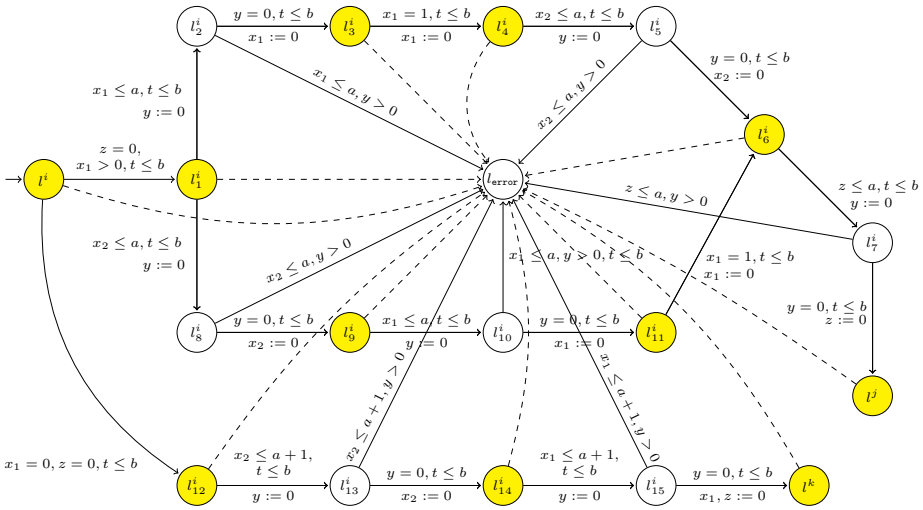


Fig. 3: decrement gadget

$(l^i, (w(x_1), w(x_2), 0, w(y)) \xrightarrow{0} (l^i_1, (w(x_1), w(x_2), 0, w(y)) \xrightarrow{v(a)-w(x_1)} (l^i_2, (v(a), w(x_2)+v(a)-w(x_1), v(a)-w(x_1), 0)) \xrightarrow{0} (l^i_3, (0, w(x_2)+v(a)-w(x_1), v(a)-w(x_1), 0)) \xrightarrow{1} (l^i_4, (0, w(x_2)+v(a)-w(x_1)+1, v(a)-w(x_1)+1, 1)) \xrightarrow{w(x_1)-w(x_2)-1} (l^i_5, (w(x_1)-w(x_2)-1, v(a), v(a)-w(x_2), 0)) \xrightarrow{0} (l^i_6, (w(x_1)-w(x_2)-1, 0, v(a)-w(x_2), 0)) \xrightarrow{w(x_2)} (l^i_7, (w(x_1)-1, w(x_2), v(a), 0)) \xrightarrow{0} (l^j, (w(x_1)-1, w(x_2), 0, 0))$.

Simulating the 2-counter machine Now, consider the runs $\rho_{\spadesuit}$ that take a u-guard $x \leq a$ "too early". At this moment, since after a small amount of time we have $x \leq a$ and $y > 0$ are true, there is a run that eventually reaches $l_{\texttt{error}}$ and can never leave it; hence $\mathsf{EGAF}_{=0}\heartsuit$ does not hold for these runs. The same way, the runs $\rho_{\spadesuit}$ that take an unguarded transition to $l_{\texttt{error}}$ (whether or not $t \leq b$ is true) are stuck in a location labeled by $\spadesuit$; hence $\mathsf{EGAF}_{=0}\heartsuit$ does not hold for these runs. In the following, we do not consider these runs anymore.

Now, let us consider the runs $\rho_{\heartsuit}$ that take each u-guard at the very last moment, which is exactly when a clock $w(x) = v(a)$.

- If the two-counter machine halts then, there exist parameter valuations v (typically $v(a)$ larger than the maximum value of the counters during the computation and $v(b)$ larger than the duration of the corresponding run in $\mathcal{A}$), for which there is a (unique) run in the constructed U-PTA simulating correctly the machine, reaching $l_{\texttt{halt}}$ and staying there forever, so $\mathsf{EGAF}_{=0}\heartsuit$ holds for these valuations: hence $\mathsf{EGAF}_{=0}\heartsuit$-emptiness is false.
- Conversely, if the two-counter machine does not halt, then for any valuation, all runs either end in $l_{\texttt{error}}$ (either because they took an unguarded transition

to $l_{\texttt{error}}$ or because they blocked due to the guard $t \leq b$—each gadget takes at least one time unit, so we can combine at most $v(b)$ gadgets—and again reached $l_{\texttt{error}}$); hence there is no parameter valuation for which $\mathsf{EGAF}_{=0}\heartsuit$ holds. Then $\mathsf{EGAF}_{=0}\heartsuit$-emptiness is true.

Therefore $\mathsf{EGAF}_{=0}\heartsuit$-emptiness is true iff the two-counter machine does not halt. □

Remark 1 (CTL). We may wonder if the *timed* aspect of TCTL is responsible for the undecidability. In fact, it is not, and we could modify the proof to show that CTL itself leads to undecidability. The idea is that we remove the unguarded transitions in both the increment and the decrement and 0-test gadgets, label each location of $L \setminus \{l_{\texttt{error}}\}$ with $\heartsuit$, and add an unguarded self-loop on $l_{\texttt{halt}}$. We claim that EGAX-emptiness is undecidable: we show that $\mathsf{EGAX}\heartsuit$ holds for a unique run of a U-PTA that simulates a two-counter machine, with a similar reasoning.

4 Undecidability for bounded U-PTAs

We now show that undecidability remains even when the parameter domain is bounded. Note that, if we were addressing the full class of PTAs, showing an undecidability result for bounded PTAs automatically extends to the full class of PTAs, as we can simulate any bounded PTA by an unbounded PTA (see, e. g., [ALR16b, Fig. 3]). This is not the case for U-PTAs: indeed, in [ALR16b], we showed that bounded (L/)U-PTAs are incomparable with (L/)U-PTAs; that is, it is impossible to simulate a bounded U-PTA using a U-PTA (e. g., by using a gadget that enforces parameters to be bounded), due to the nature of guards, preventing us to artificially bound a parameter both from above and from below (in fact, for U-PTAs, bounding from below is possible, but not from above). Therefore, we must study both problems. Finally note that the EG-emptiness is decidable for bounded L/U-PTAs but undecidable for L/U-PTAs [AL17], which motivates further the need to investigate both versions.

Theorem 2. *The $\mathsf{EGAF}_{=0}$-emptiness is undecidable for bounded U-PTAs.*

We reduce this time from the boundedness problem for two-counter machines (i. e., whether the value of the counters remains bounded along the execution), which is undecidable [KC10].

We define a U-PTA that, under some conditions, will encode the machine, and for which $\mathsf{EGAF}_{=0}\heartsuit$-emptiness holds iff the counters in the machine remain bounded. The idea is as follows: we reuse a different encoding (originally from [ALR16a]), and apply the same modifications as we did in the proof of Theorem 1.

Our U-PTA $\mathcal{A}$ uses one parameter a, and four clocks i. e., a single non-parametric clock y and three parametric clocks x_1, x_2, z. Each state q_i of the two-counter machine is encoded by a location l^i of $\mathcal{A}$. Each increment instruction

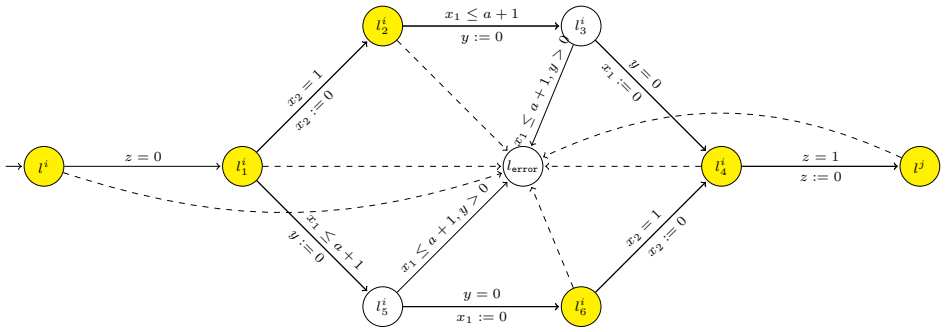


Fig. 4: increment gadget

of the two-counter machine is encoded into a U-PTA fragment. The decrement instruction is a modification of the one in [ALR16a] using the same modifications as the increment gadget.

Given v, our encoding is such that when in l^i with $w(z) = 0$ then $w(x_1)$ (resp. $w(x_2)$) represents the value of the counter $\mathtt{c}_1$ (resp. $\mathtt{c}_2$) encoded by $1 - v(a)\mathtt{c}_1$ (resp. $1-v(a)\mathtt{c}_2$). Each of our locations has exactly one label: ♡ for the locations already present in [ALR16a] (depicted in yellow in our figures), and ♠ for the newly introduced locations (depicted in white in our figures).

We assume $a \in [0, 1]$. The initial encoding when $w(z) = 0$ is $w(x_1) = 1 - v(a)\mathtt{c}_1, w(x_2) = 1-v(a)\mathtt{c}_2, w(y) = 0$. Suppose $w(x_2) \leq w(x_1)$. From l^i, we prove that there is a unique run, going through the upper branch of the gadget, that reaches l^j without violating our property. It is the one that takes each transition with a u-guard $x \leq a + 1$ at the exact moment $w(x) = v(a) + 1$:

$(l^i, w) \xrightarrow{0} (l^i_1, (1 - v(a)\mathtt{c}_1, 1 - v(a)\mathtt{c}_2, 0, 0)) \xrightarrow{v(a)\mathtt{c}_2} (l^i_2, (1 - v(a)\mathtt{c}_1 + v(a)\mathtt{c}_2, 0, v(a)\mathtt{c}_2, v(a)\mathtt{c}_2)) \xrightarrow{v(a)-v(a)\mathtt{c}_2+v(a)\mathtt{c}_1} (l^i_3, (v(a)+1, v(a)-v(a)\mathtt{c}_2+v(a)\mathtt{c}_1, v(a)+v(a)\mathtt{c}_1, 0) \xrightarrow{0} (l^i_4, (0, v(a) - v(a)\mathtt{c}_2 + v(a)\mathtt{c}_1, v(a) + v(a)\mathtt{c}_1, 0) \xrightarrow{1-v(a)-v(a)\mathtt{c}_1} (l^j, (1 - v(a)(\mathtt{c}_1 + 1), 1 - v(a)\mathtt{c}_2, 0, 1 - v(a)(\mathtt{c}_1 + 1))$

The case were $w(x_2) \leq w(x_1)$ is similar, taking the lower branch of the gadget.

Now, let us consider the runs $\rho_\heartsuit$ that take each u-guard at the very last moment, which is exactly when a clock $w(x) = v(a) + 1$. (For the same reason as in the proof of Theorem 1, other runs violate the property anyway.)

– If the counters of the two-counter machine remain bounded then,
  - either the two-counter machine halts (by reaching q_{halt}) and there exist parameter valuations v (typically $v(a)$ small enough to encode the required value of the counters during the computation), for which there is a (unique) run in the constructed U-PTA simulating correctly the machine, reaching $l_{\mathtt{halt}}$ and staying there forever, so $\mathsf{EGAF}_{=0}\heartsuit$ holds for these valuations: hence $\mathsf{EGAF}_{=0}\heartsuit$-emptiness is false;

- or the two-counter machine loops forever (and never reaches q_{halt}) with bounded values of the counters, and again there exist parameter valuations v (again small enough to encode the maximal value of the counters) for which there is an infinite (unique) run in the U-PTA simulating correctly the machine. As this run is infinite, we infinitely often visit the decrement and/or the increment gadget(s), so $\mathsf{EGAF}_{=0}\heartsuit$ holds for these valuations: hence $\mathsf{EGAF}_{=0}\heartsuit$-emptiness is again false.

– Conversely, if the counters of the two-counter machine are unbounded, then for any valuation, all runs either end in $l_{\texttt{error}}$, either because they took an unguarded transition to $l_{\texttt{error}}$ or because they blocked due to the guard $x \leq a+1$ —indeed when in l_6^i, we have $w(z) = v(a)(\mathsf{c}_1+1)$ so if c_1 is unbounded, after a sufficient number of steps we cannot pass the guard $z = 1$— and again reached $l_{\texttt{error}}$. Hence there is no parameter valuation for which $\mathsf{EGAF}_{=0}\heartsuit$ holds. Then $\mathsf{EGAF}_{=0}\heartsuit$-emptiness is true.

Using the same reasoning as in the proof of Theorem 1 and [ALR16a], we conclude that $\mathsf{EGAF}_{=0}\heartsuit$-emptiness is true iff the values of the counters of the two-counter machine are unbounded.

5 Decidability of flat-TCTL for L/U-PTAs without invariants

In this section, we prove that the EG-emptiness and universality problems are decidable for L/U-PTAs without invariants and with integer-valued parameters. Recall that for L/U-PTAs in their classical form with invariants (even over integer-valued parameters), these same problems are undecidable [AL17]. 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. As our definition of L/U-PTAs does not involve invariants, our model is a subclass of L/U-PTAs as defined in [HRSV02,BL09]. Therefore, it holds for our definition of L/U-PTAs.

Lemma 1 (monotonicity). *Let $\mathcal{A}$ be an L/U-PTA without invariant 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 $v'(\mathcal{A})$.*

We will see that EG-emptiness can be reduced to the following two problems. The first one is *cycle-existence* [AL17]: given a TA $v(\mathcal{A})$, is there at least one run of $v(\mathcal{A})$ with an infinite number of discrete transitions? Before introducing the second problem, we need to have a closer look at deadlocks: recall that a state is deadlocked when no discrete transition can be taken, even after elapsing some time. As we do not have invariants, it will be either a state with no outgoing edge, or a state in which each outgoing transition contains at least one constraint on any clock x of the form $x \triangleleft k$, where k is a constant, or $x \triangleleft p^+$, where p^+ is a

parameter. Indeed, for any parameter valuation, it suffices to wait enough time until all such guards are disabled—and the state becomes deadlocked. Note that with invariants, like in the L/U-PTAs of [HRSV02], this would not be sufficient: a state containing an invariant $x \triangleleft k$ and a transition containing a constraint $x \triangleleft k$ is not a deadlocked state, as the transition is forced to be taken. Formally, given an L/U-PTA[3] $\mathcal{A} = (\Sigma, L, \mathbf{L}, l_0, \mathbb{X}, \mathbb{P}, E)$, we define $L_D(\mathcal{A}) := \{l \in L \mid$ for all edges $(l, g, a, R, l') \in E$, g contains at least one constraint on a clock x of the form $x \triangleleft k$, where $k \in \mathbb{N}$, or $x \triangleleft p^+$, where $p^+ \in \mathbb{P}\}$.[4]

Now, the second problem we need to distinguish is *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)? As mentioned above, unlike the L/U-PTAs of [HRSV02], given an L/U-PTA $\mathcal{A}$, detecting deadlocks is equivalent in our L/U-PTAs without invariants to the *reachability* problem of a given location of $L_D(\mathcal{A})$. Let $v_{0/\infty}$ be the parameter valuation s.t. for each lower-bound parameter p^-, $v_{0/\infty}(p^-) = 0$ and for each upper-bound parameter p^+, $v_{0/\infty}(p^+) = \infty$.

Recall that $\mathsf{EG}\, T$ holds if either there is an infinite run staying in T, or there is a finite deadlocked run staying in T.

Lemma 2. *Let $\mathcal{A}$ be an L/U-PTA without invariant. There is a deadlock in $v(\mathcal{A})$ for some parameter valuation v iff there is $l \in L_D(\mathcal{A})$ reachable in $v_{0/\infty}(\mathcal{A})$.*

Proof. $\Rightarrow$ Suppose $v(\mathcal{A})$ is deadlocked. There is a run in $v(\mathcal{A})$ ending in a state (l, w) with no possible outgoing transition. That means for all edges $(l, g, a, R, l') \in E$, guard $v(g)$ is not satisfied by $w + d$, for all $d \geq 0$. In particular, let M be the maximal constant appearing in the guards of $v_{0/\infty}(\mathcal{A})$ plus one, then g is not satisfied for $w + M$. Yet, for that clock valuation, for sure, all simple constraints of the form $k \triangleleft x$ are satisfied, so this means that g must contain at least one constraint on a clock x of the form $x \triangleleft k$, where $k \in \mathbb{N}$ and $k < w(x) + M$, or $x \triangleleft p^+$, where $p^+ \in \mathbb{P}$ and $v(p^+) < w(x) + M$. Therefore, $l \in L_D(\mathcal{A})$.

Moreover as constraints in $v(\mathcal{A})$ are stronger than those in $v_{0/\infty}(\mathcal{A})$ (i. e., for each lower-bound parameter p^-, $v_{0/\infty}(p^-) \leq v(p^-)$ and for each upper-bound parameter p^+, $v(p^+) \leq v_{0/\infty}(p^+)$), from Lemma 1 l is reachable along a run of $v_{0/\infty}(\mathcal{A})$.

$\Leftarrow$ Conversely, let $l \in L_D(\mathcal{A})$ and suppose there is a run of $v_{0/\infty}(\mathcal{A})$ reaching (l, w), for some clock valuation w. Let v be the parameter valuation, defined as in the proof of [HRSV02, Prop 4.4], such that (l, w) is also reachable in $v(\mathcal{A})$. That valuation assigns a finite value to upper bound parameters that we denote by μ.

Let $e = (l, g, a, R, l') \in E$. For each constraint of the form $x \triangleleft k$ with $k \in \mathbb{N}$ in g, define $d_1 = \max(0, \max_x(k - w(x))) + 1$. Then, for all clocks x and for all $d \geq d_1$, $w(x) + d \triangleleft k$ is false. Similarly, for each constraint of the form $x \triangleleft$

[3] Throughout this section, we do not use the labeling function $\mathbf{L}$.

[4] Observe that this definition also includes the locations with syntactically no outgoing edge at all.

p^+ with p^+ an upper-bound parameter in g, define $d_2 = \max(0, \max_x(\mu - w(x))) + 1$. Then, for all clocks x and for all $d \geq d_2$, $w(x) + d \triangleleft v(p^+)$ is false. Let $d_0 = \max(d_1, d_2)$ then, by construction $(l, w + d_0)$ is a deadlocked state in $v(\mathcal{A})$. □

Consider now a TA without invariants $\mathcal{A}$, and a subset T of its locations. We build a TA $T^+(\mathcal{A})$ as follows: first remove all locations not in T and remove all transitions to and from those removed locations. Second, add self-loops to all locations in $L_D(\mathcal{A})$, with a guard that is true, and no reset.

Lemma 3. $\mathsf{EG}(T)$ *holds if and only if there exists an infinite run in* $T^+(\mathcal{A})$.

Proof. $\Rightarrow$ Suppose $\mathsf{EG}(T)$ holds. Then there is a maximal path in $\mathcal{A}$ that stays in T. If that path is infinite then, by construction it is still possible in $T^+(\mathcal{A})$. Otherwise, it is finite and therefore it is a deadlock. From Lemma 2, this means that some location in $T \cap L_D(\mathcal{A})$ is reachable in $\mathcal{A}$, by always staying in T. Consequently that location is still reachable in $T^+(\mathcal{A})$ and since it belongs to $L_D(\mathcal{A})$, it has a self-loop in $T^+(\mathcal{A})$, which implies that there is an infinite run there.

$\Leftarrow$ In the other direction, suppose that there is an infinite run in $T^+(\mathcal{A})$. Either the corresponding infinite path never uses any of the added self-loops and therefore it is possible as is in $\mathcal{A}$, which implies $\mathsf{EG}(T)$, or it goes through $L_D(\mathcal{A})$ at least once. The latter means that some location in $L_D(\mathcal{A})$ is reachable in $\mathcal{A}$ by staying in T, and by Lemma 2, this implies that there exists a finite maximal path in $\mathcal{A}$, and finally that we have $\mathsf{EG}(T)$ in $\mathcal{A}$. □

Corollary 1. *The* EG*-emptiness and* EG*-universality problems are PSPACE-complete for integer-valued L/U-PTAs without invariants.*

Proof. PSPACE-hardness comes from the fact that an L/U-PTA that does not use parameters in guards is a TA and EG is PSPACE-hard for TAs [AD94].

Let $\mathcal{A}$ be an L/U-PTA and T a subset of its locations. Remark that the construction of Lemma 3 is independant of the constants in the guards, and hence can be done in the same way for a PTA, giving another PTA $T^+(\mathcal{A})$ such that, for all parameter valuations v, $T^+(v(\mathcal{A})) = v(T^+(\mathcal{A}))$. By Lemma 3, EG-emptiness (resp. EG-universality) then reduces to the emptiness (resp. universality) of the set of parameter valuations v such that $v(T^+(\mathcal{A}))$ has an infinite accepting path. We conclude by recalling that the latter problem can be solved in PSPACE for both emptiness and universality [BL09]. □

This result is important as it is the first non-trivial subclass of PTAs for which EG-universality (equivalent by negation to AF-emptiness) is decidable.

We already had the same complexity for EF-emptiness and EF-universality [HRSV02], and by negation we can get the other flat formulas of TCTL, both for universality and emptiness (e. g., AF-emptiness is "not EG-universality"). It is also easy to see that all those results would hold for flat formulas using the "until" operator. Therefore we have:

Theorem 3. *Flat-TCTL-emptiness and flat-TCTL-universality are PSPACE-complete for integer-valued L/U-PTAs without invariant.*

Remark 2. These results come without Flat-TCTL-synthesis. Indeed, suppose we can compute the set of parameters s.t. a Flat-TCTL formula is satisfied by an integer-valued L/U-PTAs without invariant, say EF, and check for the emptiness of its intersection with a set of equality constraints. Consider an integer-valued PTA $\mathcal{A}$ without invariants. For each parameter p of $\mathcal{A}$ that is used both as an upper-bound and as a lower-bound, syntactically replace its occurrences as an upper-bound (resp. lower-bound) by a new parameter p^+ (resp. p^-). We obtain an integer-valued L/U-PTAs without invariant $\mathcal{A}'$. By hypothesis, let S be the solution set of parameters valuations to the EF-synthesis problem for $\mathcal{A}'$. Let S' be the set of equality constraints $p^+ = p^-$. Therefore we can decide whether $S \cap S' = \emptyset$ and the EF-emptiness problem is decidable for integer-valued PTAs without invariants, in contradiction with the results of [BBLS15].

6 Conclusion and perspectives

In this paper, we solved the open problem of the nested TCTL-emptiness for U-PTAs, that implies the undecidability of the whole TCTL-emptiness problem for this subclass of L/U-PTAs. Note that our proof holds even for integer-valued parameters, and even without invariants. This is a reminder that the border between undecidability and decidability problems for L/U-PTAs and its subclasses is quite thin. We used a reduction from a U-PTA to a two-counter machine using several gadgets to prove that a precise TCTL-emptiness problem is undecidable. Unlike PTAs and bounded PTAs, U-PTAs and bounded U-PTAs are incomparable, hence we had to verify whether the same reasoning was applicable when the parameter domain is bounded. For this purpose, we used another construction to reduce to a bounded U-PTA from a two-counter machine to prove that the same TCTL-emptiness problem is also undecidable.

Moreover, we proved that EG-emptiness and universality are PSPACE-complete for (unbounded) integer-valued L/U-PTAs without invariants. This result is particularly interesting as it was undecidable with invariants [AL17]. Using existing results, we have that flat TCTL-emptiness and universality are decidable for this class, and therefore for integer-valued U-PTAs without invariants, which contrasts with our undecidability result and shows that we are there again at the frontier of decidability.

Future work This work opens new perspectives: where exactly the undecidability starts (in particular whether EG and AF are decidable for U-PTAs with invariants or real-valued parameters, which remains open, see Table 1), whether our proofs in Sections 3 and 4 can be extended over bounded time, and whether the same results hold for L-PTAs (lower-bound PTAs).

Also, extending our decidability result in Theorem 3 while keeping decidability will be an interesting challenge.

References

ACD93. Rajeev Alur, Costas Courcoubetis, and David Dill. Model-checking in dense real-time. *Information and Computation*, 104(1):2–34, 1993.

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

AHV93. Rajeev Alur, Thomas A. Henzinger, and Moshe Y. Vardi. Parametric real-time reasoning. In S. Rao Kosaraju, David S. Johnson, and Alok Aggarwal, editors, *STOC*, pages 592–601, New York, NY, USA, 1993. ACM.

AL17. Étienne André and Didier Lime. Liveness in L/U-parametric timed automata. In *ACSD*, pages 9–18. IEEE, 2017.

ALR16a. Étienne André, Didier Lime, and Olivier H. Roux. Decision problems for parametric timed automata. In Kazuhiro Ogata, Mark Lawford, and Shaoying Liu, editors, *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.

And17. Étienne André. What's decidable about parametric timed automata? *International Journal on Software Tools for Technology Transfer*, 2017. To appear.

BBLS15. Nikola Beneš, Peter Bezděk, Kim Gulstrand 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 Erzsébet Csuhaj-Varjú, Martin Dietzfelbinger, and Zoltán Ésik, editors, *MFCS, Part I*, volume 8634 of *Lecture Notes in Computer Science*, pages 123–134. Springer, 2014.

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.

JLR13. Aleksandra Jovanović, Didier Lime, and Olivier H. Roux. Integer parameter synthesis for timed automata. In *TACAS*, volume 7795 of *Lecture Notes in Computer Science*, pages 401–415. Springer, 2013.

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.

KC10. E. V. Kuzmin and D. J. Chalyy. Decidability of boundedness problems for minsky counter machines. *Automatic Control and Computer Sciences*, 44(7):387–397, 2010.

Mil00. Joseph S. Miller. Decidability and complexity results for timed automata and semi-linear hybrid automata. In Nancy A. Lynch and Bruce H. Krogh, editors, *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., Upper Saddle River, NJ, USA, 1967.