Form Methods Syst Des (2009) 35: 121–151
DOI 10.1007/s10703-009-0074-0

Decision problems for lower/upper bound parametric timed automata

Laura Bozzelli · Salvatore La Torre

Received: 21 September 2008 / Accepted: 28 May 2009 / Published online: 9 June 2009

Abstract We investigate a class of parametric timed automata, called lower bound/upper bound (L/U) automata, where each parameter occurs in the timing constraints either as a lower bound or as an upper bound. For such automata, we show that basic decision problems, such as emptiness, finiteness and universality of the set of parameter valuations for which there is a corresponding infinite accepting run of the automaton, is PSPACE-complete. We extend these results by allowing the specification of constraints on parameters as a linear system. We show that the considered decision problems are still PSPACE-complete, if the lower bound parameters are not compared with the upper bound parameters in the linear system, and are undecidable in general. Finally, we consider a parametric extension of $\mathsf{MITL}_{0,\infty}$, and prove that the related satisfiability and model checking (w.r.t. L/U automata) problems are PSPACE-complete.

Keywords Parametric real-time verification · Parametric timed automata · Timed temporal logics · Synthesis of parameters

1 Introduction

Timed automata [1] are a widely accepted formalism to model the behavior of real-time systems. A timed automaton is a finite–state transition graph equipped with a finite set of *clock variables* which are used to express *timing constraints*. The semantics is given by an infinite-state transition system where transitions correspond either to a change of location (instantaneous transition) or to a time consumption (time transition). Over the years, timed

This research was partially supported by the MIUR grants ex-60% 2006 and 2007 Università di Salerno.
A preliminary version of this paper appears in the Proceedings of the 34th International Colloquium on Automata, Languages and Programming (ICALP'07), LNCS 4596, pp 925–936. Springer, 2007.

L. Bozzelli (✉)
Università dell'Insubria, Via Valleggio 11, 22100 Como, Italy
e-mail: laura.bozzelli@dma.unina.it

S. La Torre
Università di Salerno, Via Ponte Don Melillo, 84084 Fisciano, Italy
e-mail: latorre@dia.unisa.it

automata have been intensively studied by many authors, and significant progresses have been done in developing verification algorithms, heuristics, and tools (see [2] for a recent survey).

Timing constraints in timed automata allow the specification of constant bounds on delays among events. Typical examples are upper and lower bounds on computation times, message delays and timeouts. In the early stages of a design, when not much is known about the system under development, it is however useful for designers to use parameters instead of specific constants.

In [4], Alur et al. introduce parametric timed automata, i.e., timed automata where clocks can be compared to parameters. For such class of automata, they study the *emptiness problem*: "is there a parameter valuation for which the automaton has an accepting run?" This problem turns out to be undecidable already for parametric timed automata with only three parametric clocks, while it is decidable when at most one clock is compared to parameters. In case of two parametric clocks, the emptiness problem is closely related to various hard and open problems of logic and automata theory [4]. In [13], Hune et al. identify a subclass of parametric timed automata, called *lower bound/upper bound* (*L/U*) *automata*, in which each parameter occurs either as a lower bound or as an upper bound in the timing constraints. Despite this limitation, the model is still interesting in practice. In fact, L/U automata can be used to model the Fisher's mutual exclusion algorithm [15], the root contention protocol [14], and other known examples from the literature (see [13]). Hune et al. show that the emptiness problem for L/U automata with respect to finite runs is decidable. The case of *infinite* accepting runs (which is crucial for the verification of liveness properties) is not investigated, and does not follow from their results.

Our contribution In this paper, we further investigate the class of L/U automata and consider acceptance conditions over infinite runs. Given an L/U automaton $\mathcal{A}$, denote with $\Gamma(\mathcal{A})$ the set of parameter valuations for which the automaton has an infinite accepting run. We show that questions about $\Gamma(\mathcal{A})$ can be answered considering a bounded set of parameter valuations of size exponential in the size of the constants and the number of clocks, and polynomial in the number of parameters and locations of $\mathcal{A}$. Therefore, we are able to show that checking the set $\Gamma(\mathcal{A})$ for emptiness, universality (i.e., if $\Gamma(\mathcal{A})$ contains all the parameter valuations), and finiteness is PSPACE-complete. The main argument for such results is as follows: suppose that $\mathcal{A}$ is an L/U automaton which uses parameters only as lower bounds (resp., upper bounds); then if an infinite run ρ is accepted by $\mathcal{A}$ for large-enough values of the parameters, we can determine appropriate finite portions of ρ which can be "repeatedly simulated" (resp., "deleted") thus obtaining a run ρ' which is accepted by $\mathcal{A}$ for larger (resp., smaller) parameters values.

Parameters in system models can be naturally related by linear equations and inequalities. As an extension of the above results, we consider *constrained emptiness* and *constrained universality* on L/U automata, where the constraint is represented by a linear system over parameters. We show that these problems are in general undecidable, and become decidable in polynomial space (and thus PSPACE-complete) if we do not compare parameters of different types in the linear constraints.

Moreover, we show that when all the parameters in the model are of the same type (i.e., either lower bound or upper bound), it is possible to compute an explicit representation of the set $\Gamma(\mathcal{A})$ by linear systems over parameters whose size is *doubly exponential* in the number of parameters.

An important consequence of our results on L/U automata is the extension to the dense-time paradigm of the results shown in [6]. We define a parametric extension of the dense-time linear temporal logic $\mathsf{MITL}_{0,\infty}$ [5], denoted $\mathsf{PMITL}_{0,\infty}$, and show that (under restrictions on the use of parameters analogous to those imposed on L/U automata) the related satisfiability and model-checking problems are PSPACE-complete. The proof consists of translating formulas to L/U automata. To the best of our knowledge this is the first work that solves verification problems against linear-time specifications with parameters both in the model and in the specification.

Related work Besides the already mentioned research, there are several other papers that are related to ours. The idea of restricting the use of parameters (in order to obtain decidability) such that upper and lower bounds cannot share a same parameter is also present in [6] where the authors study the logic LTL [16] augmented with parameters. The general structure of our argument for showing decidability ("pumping" argument) is inspired to their approach. However, let us stress that there are substantial technical differences with that paper since we consider a different framework, and in particular, we deal with a dense-time semantics. Parametric branching time specifications were first investigated in [11, 18] where decidability is shown for logics obtained as extensions of TCTL [3] with parameters. In [9], decidability is extended to full TCTL with Presburger constraints over parameters. In [8], decidability is established for the model checking problem of *discrete-time* timed automata with *one parametric clock* against parametric TCTL without equality (for full TCTL with parameters the problem is undecidable). Finally, recall that the undecidability of systems with parameters is also captured by the undecidability results shown in [12]. However, the limitations we consider for obtaining decidability seem to be orthogonal to those considered there. We are not aware of any way of obtaining our decidability results from those presented in [12].

2 Parametric timed automata and L/U automata

Throughout this paper, we fix a finite set of *parameters* $P = \{p_1, \ldots, p_m\}$. Let $\mathbb{R}_{\geq 0}$ be the set of non-negative reals, $\mathbb{N}$ the set of natural numbers, and $\mathbb{Z}$ the set of integers.

A *linear expression* e is an expression of the form $c_0 + c_1 p_1 + \cdots + c_m p_m$ with $c_0, c_1, \ldots, c_m \in \mathbb{Z}$. We say that parameter p_i *occurs* in e if $c_i \neq 0$. A *parameter valuation* is a function $v : P \to \mathbb{N}$ assigning a natural number to each parameter. For the linear expression e above, $e[v]$ denotes the integer $c_0 + c_1 v(p_1) + \cdots + c_m v(p_m)$. The *null parameter valuation*, denoted v_{null}, maps every parameter to 0. For two parameter valuations v_1 and v_2, we write $v_1 \leq v_2$ to mean that $v_1(p) \leq v_2(p)$ for all $p \in P$. Given a set Γ of parameter valuations, we say that Γ is *downward-closed* iff for all $v, v' \in \Gamma$, ($v \leq v'$ and $v' \in \Gamma$) implies $v \in \Gamma$. Analogously, we say that Γ is *upward-closed* iff for all $v, v' \in \Gamma$, ($v \leq v'$ and $v \in \Gamma$) implies $v' \in \Gamma$.

We fix a finite set of *clock* variables X. In order to obtain a more uniform notation for specifying constraints on clocks, we allow in our model a special clock $x_0 \in X$, called *zero clock*, which always evaluates to 0 (i.e., it does not increase with time).

An *atomic clock constraint* f is an expression of the form $x - y \prec e$, where $x, y \in X$, e is a linear expression, and $\prec \in \{<, \leq\}$. We say that f is *parametric* if some parameter occurs in e. A *clock constraint* is a finite conjunction of atomic constraints. A *clock valuation* is a function $w : X \to \mathbb{R}_{\geq 0}$ assigning a value in $\mathbb{R}_{\geq 0}$ to each clock and such that $w(x_0) = 0$. For a clock constraint f, a parameter valuation v, and a clock valuation w, the pair (v, w) *satisfies* f, denoted $(v, w) \models f$, if the expression obtained from f by replacing each parameter p with $v(p)$ and each clock x with $w(x)$ evaluates to true.

A *reset set* r is a subset of X containing the clocks to be reset to 0. For $\tau \in \mathbb{R}_{\geq 0}$ and a clock valuation w, the clock valuation $w + \tau$ is defined as $(w + \tau)(x) = w(x) + \tau$ for all $x \in X \setminus \{x_0\}$ and $(w + \tau)(x_0) = 0$. For a reset set $r \in 2^X$, the clock valuation $w[r]$ is defined as $w[r](x) = 0$ if $x \in r$ and $w[r](x) = w(x)$ otherwise. Let Ξ be the set of all clock constraints over X and P.

Definition 1 A *parametric timed automaton (*PTA *)* is a tuple $\mathcal{A} = \langle Q, q^0, \Delta, F\rangle$, where Q is a finite set of *locations*, $q^0 \in Q$ is the *initial location*, $\Delta \subseteq Q \times \Xi \times 2^X \times Q$ is a finite *transition relation*, and $F \subseteq Q$ is a set of *accepting locations*.

Let $\mathcal{A} = \langle Q, q^0, \Delta, F\rangle$ be a PTA . A *state* of $\mathcal{A}$ is a pair (q, w) such that $q \in Q$ is a location and w is a clock valuation. The *initial state* is $(q^0, \overrightarrow{0})$, where $\overrightarrow{0}$ maps every $x \in X$ to 0. An *extended state* of $\mathcal{A}$ is a pair (s, t) such that s is a state and $t \in \mathbb{R}_{\geq 0}$ (intuitively, t represents the total time which has elapsed). A PTA $\mathcal{A}$ is called a *timed automaton* (TA, for short) if $\mathcal{A}$ does not contain occurrences of parameters. For a PTA $\mathcal{A}$ and a parameter valuation v, we denote by $\mathcal{A}_v$ the TA obtained by replacing each linear expression e of $\mathcal{A}$ by $e[v]$.

Let $\mathcal{A} = \langle Q, q^0, \Delta, F\rangle$ be a PTA and v be a parameter valuation. The *concrete semantics of $\mathcal{A}$ under v*, denoted $[\![\mathcal{A}]\!]_v$, corresponds to the semantics of the TA $\mathcal{A}_v$ and is given by the labeled transition system $\langle S, \twoheadrightarrow\rangle$ over $(\Delta \cup \{\bot\}) \times \mathbb{R}_{\geq 0}$, where S is the set of $\mathcal{A}$ states (independent on v) and the set of labeled edges $\twoheadrightarrow \subseteq S \times [(\Delta \cup \{\bot\}) \times \mathbb{R}_{\geq 0}] \times S$ (dependent on v) is defined as follows: $(q, w)\xrightarrow{\delta,\tau}\!\!\!\!\!\rightarrow(q', w')$ iff

- *either* $\delta = (q, g, r, q')$, $\tau = 0$, $(v, w) \models g$, and $w' = w[r]$ (*instantaneous transition*),
- *or* $\delta = \bot$, $q' = q$, $\tau > 0$, and $w' = w + \tau$ (*time transition*).

An *extended edge* of $[\![\mathcal{A}]\!]_v$ is of the form $(s, t)\xrightarrow{\delta,\tau}\!\!\!\!\!\rightarrow(s', t')$ such that $s\xrightarrow{\delta,\tau}\!\!\!\!\!\rightarrow s'$ is an edge of $[\![\mathcal{A}]\!]_v$ and $t' = t + \tau$.

An *infinite run* of the TA $\mathcal{A}_v$ is an infinite path $\rho = s_0\xrightarrow{\delta_0,\tau_0}\!\!\!\!\!\rightarrow s_1\xrightarrow{\delta_1,\tau_1}\!\!\!\!\!\rightarrow s_2\cdots$ of $[\![\mathcal{A}]\!]_v$ such that $\sum_{i\geq 0}\tau_i = \infty$ (*progress condition*) and for infinitely many $i \geq 0$, $\delta_i \neq \bot$ (there are infinitely many occurrences of instantaneous transitions). Moreover, ρ is *accepting* iff for infinitely many $i \geq 0$, we have that $q_i \in F$, where $s_i = (q_i, w_i)$. A *finite run* of $\mathcal{A}_v$ is a finite path $\rho = s_0\xrightarrow{\delta_0,\tau_0}\!\!\!\!\!\rightarrow s_1\cdots s_{n-1}\xrightarrow{\delta_{n-1},\tau_{n-1}}\!\!\!\!\!\rightarrow s_n$ of $[\![\mathcal{A}]\!]_v$. The finite run ρ is *accepting* if for some $0 \leq i \leq n$, we have that $q_i \in F$, where $s_i = (q_i, w_i)$. The *duration* of ρ, denoted by $\mathrm{DUR}(\rho)$, is defined as $\mathrm{DUR}(\rho) = \sum_{i=0}^{i=n-1}\tau_i$.

The notions of *extended infinite run* and *extended finite run* of $\mathcal{A}_v$ are defined analogously. For a finite or infinite run $\rho = s_0\xrightarrow{\delta_0,\tau_0}\!\!\!\!\!\rightarrow s_1\xrightarrow{\delta_1,\tau_2}\!\!\!\!\!\rightarrow s_2\cdots$ and $0 \leq i \leq j$, we denote by $\rho[i, j]$ the finite run given by $s_i\xrightarrow{\delta_i,\tau_i}\!\!\!\!\!\rightarrow\cdots\xrightarrow{\delta_{j-1},\tau_{j-1}}\!\!\!\!\!\rightarrow s_j$. The *extended run associated with ρ* is the extended run of $\mathcal{A}_v$ given by $(s_0, t_0)\xrightarrow{\delta_0,\tau_0}\!\!\!\!\!\rightarrow(s_1, t_1)\xrightarrow{\delta_1,\tau_2}\!\!\!\!\!\rightarrow(s_2, t_2)\cdots$, where $t_0 = 0$ and $t_{i+1} = t_i + \tau_i$ for any i. Note that $t_i = \mathrm{DUR}(\rho[0, i])$ for any i.

We denote with $\Gamma(\mathcal{A})$ the set of parameter valuations v such that there exists an accepting infinite run of $\mathcal{A}_v$ starting from the initial state $(q^0, \overrightarrow{0})$. Simple questions about the set $\Gamma(\mathcal{A})$, such as emptiness, are known to be undecidable for the whole class of PTA [4]. Therefore, in this paper we focus on a subclass of PTA , called *lower bound/upper bound* automata, introduced in [13]. In order to define this subclass of parametric timed automata, we need additional definitions.

Given a linear expression $e = c_0 + c_1 p_1 + \cdots + c_m p_m$ and $p_i \in P$, we say that p_i *occurs positively in e* if $c_i > 0$. Analogously, we say that p_i *occurs negatively in e* if $c_i < 0$. A *lower bound parameter* of a PTA $\mathcal{A}$ is a parameter that if it occurs in a linear expression of $\mathcal{A}$ then

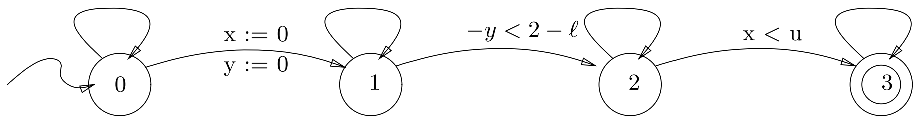

Fig. 1 An L/U automaton

it occurs negatively. Analogously, an *upper bound parameter* of a PTA $\mathcal{A}$ is a parameter that if it occurs in a linear expression of $\mathcal{A}$ then it occurs positively. We call $\mathcal{A}$ a *lower bound/upper bound* (L/U) *automaton* if every parameter occurring in $\mathcal{A}$ is either an upper bound parameter or a lower bound parameter. Moreover, we say that $\mathcal{A}$ is a *lower bound automaton* (resp., *upper bound automaton*) iff every parameter occurring in $\mathcal{A}$ is a lower bound parameter (resp., an upper bound parameter).

Example 1 Consider the automaton $\mathcal{A}$ in Fig. 1 (the reset of a clock x is denoted as $x := 0$). It has four locations $0, 1, 2, 3$, two clocks x, y, and two parameters ℓ and u. Note that the constraint $-y < 2 - \ell$ imposes a lower bound on the possible values of y, while $x < u$ imposes an upper bound on the possible values of x. Thus, ℓ and u are respectively a lower bound and an upper bound parameter, and $\mathcal{A}$ is an L/U automaton. Also, it easily follows that $\mathcal{A}_v$ has an infinite run from $(0, \overrightarrow{0})$ visiting infinitely often location 3 iff $v(\ell) < v(u) + 2$ and $v(u) > 0$. Hence, $\Gamma(\mathcal{A}) = \{v \mid v(\ell) < v(u) + 2 \text{ and } v(u) > 0\}$.

For L/U automata $\mathcal{A}$, we study some questions on the set $\Gamma(\mathcal{A})$. In particular, we consider the following decision problems:

- *Emptiness*: is the set $\Gamma(\mathcal{A})$ empty?
- *Universality*: does the set $\Gamma(\mathcal{A})$ contain all parameter valuations?
- *Finiteness*: is the set $\Gamma(\mathcal{A})$ finite?

3 Relations over states of PTA

In this section, we recall the notion of *region equivalence* in timed automata [1] and establish some basic properties of parametric timed automata which will be used to analyze the behavior of L/U automata.

For $t \in \mathbb{R}_{\geq 0}$, $\lfloor t \rfloor$ denotes the integral part of t and $\mathrm{fract}(t)$ denotes its fractional part. We define the following equivalence relations over $\mathbb{R}_{\geq 0}$:

- $t \equiv t'$ iff (i) $\lfloor t \rfloor = \lfloor t' \rfloor$ and (ii) $\mathrm{fract}(t) = 0$ iff $\mathrm{fract}(t') = 0$;
- for every $K \in \mathbb{N}$, $t \equiv_K t'$ iff either $t \equiv t'$ or $t, t' > K$.

Let $\mathcal{A} = \langle Q, q^0, \Delta, F\rangle$ be a PTA and v be a parameter valuation. Let K_v be the largest $|e[v]| + 1$ such that e is a linear expression of $\mathcal{A}$. The *region equivalence of $\mathcal{A}$ with respect to v*, denoted $\approx_v$, is the equivalence relation of finite index over $\mathcal{A}$ states defined as follows: $(q, w) \approx_v (q', w')$ iff $q = q'$ and for all clocks $x, y \in X$,

- $w(x) - w(y) \geq 0$ iff $w'(x) - w'(y) \geq 0$;
- $|w(x) - w(y)| \equiv_{K_v} |w'(x) - w'(y)|$;
- $\mathrm{fract}(w(x)) \leq \mathrm{fract}(w(y))$ iff $\mathrm{fract}(w'(x)) \leq \mathrm{fract}(w'(y))$ (*ordering of fractional parts*).

A *region* of $\mathcal{A}$ with respect to v is an equivalence class induced by $\approx_v$. Recall that the number of these regions is $O(|Q| \cdot (2K_v + 2)^{2|X|^2})$ [1] (note that we consider also diagonal constraints). Moreover, $\approx_v$ is a bisimulation over $[\![\mathcal{A}]\!]_v$. Note that if $\mathcal{A}$ is a timed automaton, then the value of K_v is obviously independent on specific valuation v, and we denote it with $K_{\mathcal{A}}$. Thus, the emptiness for a timed automaton is reduced to check the existence of *accepting* infinite paths in the finite–state quotient graph induced by region equivalence (*region graph*) [1].

Theorem 1 *Checking emptiness for a timed automaton $\mathcal{A}$ is* PSPACE*-complete and can be done in time $O(|\Delta| \cdot (2K_{\mathcal{A}} + 2)^{2|X|^2})$.*

In the following, to solve the considered problems we use arguments based on the comparison of the duration of different finite runs. Thus, for technical reasons, given a parameter valuation v, we extend the region equivalence $\approx_v$ to extended states $((q, w), t)$ of $\mathcal{A}$ (where, intuitively, t represents the total time elapsed since the beginning of the run) as follows: $((q, w), t) \approx_v ((q', w'), t')$ iff (i) $(q, w) \approx_v (q', w')$, (ii) $t \equiv t'$, and (iii) for all clocks $x \in X$, $\mathrm{fract}(w(x)) \leq \mathrm{fract}(t)$ iff $\mathrm{fract}(w'(x)) \leq \mathrm{fract}(t')$.

Note that for the time-components t and t' of two extended states we require that $t \equiv t'$ (which is stronger than condition $t \equiv_{K_v} t'$).

The claims in the following proposition essentially correspond to a reformulation of classical results on timed automata (see [1]).

Proposition 1 *Let $((q, w), t) \approx_v ((q, w'), t')$. Then,*

1. *for every $r \in 2^X$, $((q, w[r]), t) \approx_v ((q, w'[r]), t')$;*
2. *for every $\tau \geq 0$, there is $\tau' \geq 0$ such that $((q, w + \tau), t + \tau) \approx_v ((q, w' + \tau'), t' + \tau')$, $\tau \equiv \tau'$, and for every clock $x \in X$, $w(x) \equiv w'(x)$ implies $w(x) + \tau \equiv w'(x) + \tau'$.*

Proof From [1], we know $(q, w[r]) \approx_v (q, w'[r])$. From the hypothesis $((q, w), t) \approx_v ((q, w'), t')$, we get that $t \equiv t'$ and for all clocks $x \in X$, $\mathrm{fract}(w(x)) \leq \mathrm{fract}(t)$ iff $\mathrm{fract}(w'(x)) \leq \mathrm{fract}(t')$. Moreover, by definition, we have that $w(x) = w[r](x)$ and $w'(x) = w'[r](x)$, for $x \notin r$, and $w[r](x) = w'[r](x) = 0$, for $x \in r$. Thus, for all clocks $x \in X$, $\mathrm{fract}(w[r](x)) \leq \mathrm{fract}(t)$ iff $\mathrm{fract}(w'[r](x)) \leq \mathrm{fract}(t')$, and therefore, Property 1 holds.

To show Property 2, it is sufficient to consider the case $0 < \tau < 1$. In fact, if τ is an integer we can choose $\tau' = \tau$ and the stated property trivially follows. If $\tau > 1$ is not an integer, we can reduce the proof to the case $0 < \tau < 1$ by simply observing that $((q, w + \lfloor \tau \rfloor), t + \lfloor \tau \rfloor) \approx_v ((q, w' + \lfloor \tau \rfloor), t' + \lfloor \tau \rfloor)$ must hold.

Fix $0 < \tau < 1$. Denote with $a_1, \ldots, a_n$ the values $w(x)$, for $x \in X \setminus \{x_0\}$, and t sorted according to their fractional parts by increasing values. Analogously, we denote with $a'_1, \ldots, a'_n$ the tuple obtained similarly from $w'(x)$, for $x \in X \setminus \{x_0\}$, and t'. Since $((q, w), t) \approx_v ((q, w'), t')$, observe that t and t' appear at the same position respectively in $a_1, \ldots, a_n$ and $a'_1, \ldots, a'_n$, and analogously for $w(x)$ and $w'(x)$ for each $x \in X \setminus \{x_0\}$.

We distinguish two cases.

If $\mathrm{fract}(a_n) + \tau < 1$, then we can choose $0 < \tau' < 1$ such that also $\mathrm{fract}(a'_n) + \tau' < 1$ (such τ' always exists since $\mathrm{fract}(a) < 1$ for any $a \geq 0$). Thus, for each $1 \leq i \leq n$, $\mathrm{fract}(a_i + \tau) = \mathrm{fract}(a_i) + \tau$ and $\mathrm{fract}(a'_i + \tau') = \mathrm{fract}(a'_i) + \tau'$, and hence, both $\mathrm{fract}(a_i + \tau) > 0$ and $\mathrm{fract}(a'_i + \tau') > 0$ hold. Also, for each $1 \leq i \leq n$, $\lfloor a_i + \tau \rfloor = \lfloor a_i \rfloor$ and $\lfloor a'_i + \tau' \rfloor = \lfloor a'_i \rfloor$. Therefore, Property 2 easily follows in this case.

In the other case, i.e., $\mathrm{fract}(a_n) + \tau \geq 1$, let k be the smallest index $1 \leq i \leq n$ such that $\mathrm{fract}(a_i) + \tau \geq 1$. We choose $\tau' > 0$ such that k is also the smallest index $1 \leq i \leq n$

such that $\mathrm{fract}(a'_i) + \tau' \geq 1$. Since $((q, w), t) \approx_v ((q, w'), t')$, we have that $\mathrm{fract}(w(x)) \leq \mathrm{fract}(w(y))$ iff $\mathrm{fract}(w'(x)) \leq \mathrm{fract}(w'(y))$ for all $x, y \in X$, $t \equiv t'$, and $w(x) \equiv_{K_v} w'(x)$ for every clock $x \in X$ (recall that $x_0 \in X$). Thus, $\mathrm{fract}(a'_k) = 0$ iff $\mathrm{fract}(a_k) = 0$ also holds, and hence we can require that the chosen τ' also satisfies both conditions $\tau' < 1$ (thus $\tau \equiv \tau'$) and $\mathrm{fract}(a_k) + \tau > 1$ iff $\mathrm{fract}(a'_k) + \tau' > 1$. As a consequence, we get:

- for all $1 \leq i, j \leq n$, $\mathrm{fract}(a_i + \tau) \leq \mathrm{fract}(a_j + \tau)$ iff $\mathrm{fract}(a'_i + \tau') \leq \mathrm{fract}(a'_j + \tau')$;
- for all $1 \leq i \leq n$, $a_i \equiv a'_i$ implies $a_i + \tau \equiv a'_i + \tau'$;
- for all $1 \leq i \leq n$, $a_i + \tau \equiv_{K_v} a'_i + \tau'$.

Therefore, we get that $t + \tau \equiv t' + \tau'$, and for all $x \in X$, if $w(x) \equiv w'(x)$ then $(w + \tau)(x) \equiv (w' + \tau')(x)$.

Moreover, observe that from $((q, w), t) \approx_v ((q, w'), t')$ we immediately obtain that: (1) $(w + \tau)(x) - (w + \tau)(x) \geq 0$ iff $(w' + \tau')(x) - (w' + \tau')(y) \geq 0$ for all $x, y \in X$, and (2) $|(w + \tau)(x) - (w + \tau)(y)| \equiv_{K_v} |(w' + \tau')(x) - (w' + \tau')(y)|$ for all $x, y \in X$.

Thus, we can conclude that $((q, w + \tau), t + \tau) \approx_v ((q, w' + \tau'), t' + \tau')$, and Property 2 holds. □

For a PTA $\mathcal{A}$, a parameter valuation v *evaluates negative* (resp., *evaluates positive*) for $\mathcal{A}$ if for each parametric atomic constraint $x - y \prec e$ of $\mathcal{A}$, $e[v] < 0$ (resp., $e[v] > 0$). We denote by $X(P, \mathcal{A})$ the set of *parametric clocks*, that is the set of $x \in X$ such that $\mathcal{A}$ contains a parametric atomic constraint of the form $x - y \prec e$ or $y - x \prec e$.

To answer questions on $\Gamma(\mathcal{A})$, for a parametric timed automaton $\mathcal{A}$, we need to examine an infinite class of region graphs, one for each parameter valuation. However, in the next section we will show that for an L/U automaton $\mathcal{A}$, it is possible to effectively determine a parameter valuation v such that our decision problems can be reduced to check emptiness of $\mathcal{A}_v$. In our arguments, we use a preorder $\sqsubseteq$ over the set of states defined as $(q, w) \sqsubseteq (q', w')$ iff

- $(q, w) \approx_{v_{null}} (q', w')$ (recall that v_{null} is the null parameter valuation);
- for all clocks $x, y \in X(P, \mathcal{A}) \cup \{x_0\}$ such that $w(x) - w(y) > 0$: either $w'(x) - w'(y) \geq w(x) - w(y)$, or $(w'(x) - w'(y)) \equiv (w(x) - w(y))$ hold.

The first condition establishes that (q, w) and (q', w') are equivalent w.r.t. all non-parametric clock constraints of the given PTA $\mathcal{A}$. The second condition ensures that for a parameter valuation v which *evaluates negative* (resp., *evaluates positive*) for $\mathcal{A}$, each parametric clock constraint of $\mathcal{A}$ which is fulfilled in (q, w) (resp., (q', w')) w.r.t. v is also fulfilled in (q', w') (resp., (q, w)) w.r.t. v.[1] We also extend $\sqsubseteq$ to extended states of the PTA $\mathcal{A}$ as follows:

$$(s, t) \sqsubseteq (s', t') \quad \text{iff } s \sqsubseteq s' \text{ and } (s, t) \approx_{v_{null}} (s', t').$$

The following proposition summarizes some basic properties of the preorder $\sqsubseteq$. In particular, Property 4 asserts that for a parameter valuation v which evaluates negative for the given PTA $\mathcal{A}$, $\sqsubseteq$ is a simulation in $[\![\mathcal{A}]\!]_v$, while Property 5 asserts that for a parameter valuation v which evaluates positive for $\mathcal{A}$, the reverse relation of $\sqsubseteq$ is a simulation in $[\![\mathcal{A}]\!]_v$. For example, assume that v is a parameter valuation which evaluates negative for a PTA $\mathcal{A}$, $\rho = s_0 \xrightarrow{\delta_0, \tau_0} s_1 \cdots s_n \xrightarrow{\delta_n, \tau_n} s_{n+1}$ is a finite run of $\mathcal{A}_v$, and s'_0 is a state such that $s_0 \sqsubseteq s'_0$. Note that $(s_0, 0) \sqsubseteq (s'_0, 0)$. Then, by repeatedly applying Property 4 to the single steps of

[1] See proofs of Properties 4 and 5 of the following Proposition 2.

the extended run of $\mathcal{A}_v$ associated with ρ, we deduce the existence of an extended run of $\mathcal{A}_v$ whose associated run has the form $\rho' = s'_0 \xrightarrow{\delta_0, \tau'_0} s'_1 \cdots s'_n \xrightarrow{\delta_n, \tau'_n} s'_{n+1}$ such that $s_i \sqsubseteq s'_i$ and $\text{DUR}(\rho'[0,i]) \equiv \text{DUR}(\rho[0,i])$ for each $0 \le i \le n+1$. We say that ρ' is a *simulation* of ρ. Note that the motivation to extend $\sqsubseteq$ to extended states is to guarantee that in the simulations ρ' of ρ, the time durations are preserved w.r.t. $\equiv$, i.e. $\text{DUR}(\rho) \equiv \text{DUR}(\rho')$.

Proposition 2 *Let $(s,t) \sqsubseteq (s',t')$ with $s = (q,w)$ and $s' = (q,w')$. Then,*

1. *For every $r \in 2^X$ and location q_1, $((q_1, w[r]), t) \sqsubseteq ((q_1, w'[r]), t')$.*
2. *For every $\tau \ge 0$, there is $\tau' \ge 0$ such that $((q, w+\tau), t+\tau) \sqsubseteq ((q, w'+\tau'), t'+\tau')$.*
3. *For every $\tau' \ge 0$, there is $\tau \ge 0$ such that $((q, w+\tau), t+\tau) \sqsubseteq ((q, w'+\tau'), t'+\tau')$.*
4. *Let $(s,t) \xrightarrow{\delta,\tau} (s_1,t_1)$ be an extended edge of $[\![\mathcal{A}]\!]_v$, where v is a parameter valuation which* evaluates negative *for $\mathcal{A}$. Then, there is an extended edge $(s',t') \xrightarrow{\delta,\tau'} (s'_1,t'_1)$ of $[\![\mathcal{A}]\!]_v$ such that $(s'_1,t'_1) \sqsupseteq (s_1,t_1)$.*
5. *Let $(s',t') \xrightarrow{\delta,\tau'} (s'_1,t'_1)$ be an extended edge of $[\![\mathcal{A}]\!]_v$, where v is a parameter valuation which* evaluates positive *for $\mathcal{A}$. Then, there is an extended edge $(s,t) \xrightarrow{\delta,\tau} (s_1,t_1)$ of $[\![\mathcal{A}]\!]_v$ such that $(s_1,t_1) \sqsubseteq (s'_1,t'_1)$.*

Proof

1. Property 1 is a direct consequence of Proposition 1(1) and definition of $\sqsubseteq$.
2. Let $\tau \in \mathbb{R}_{\ge 0}$. By Proposition 1(2), there is a $\tau' \in \mathbb{R}_{\ge 0}$ such that $\tau \equiv \tau'$, $((q, w+\tau), t+\tau) \approx_{v_{null}} ((q, w'+\tau'), t'+\tau')$, and for every clock $x \in X$, if $w(x) \equiv w'(x)$, then $w(x)+\tau \equiv w'(x)+\tau'$. We just need to show that: for all clocks $x, y \in X(P,\mathcal{A}) \cup \{x_0\}$, if $(w+\tau)(x) - (w+\tau)(y) > 0$, then either $(w'+\tau')(x) - (w'+\tau')(y) \ge (w+\tau)(x) - (w+\tau)(y)$ or $(w'+\tau')(x) - (w'+\tau')(y) \equiv (w+\tau)(x) - (w+\tau)(y)$. Assume that $(w+\tau)(x) - (w+\tau)(y) > 0$, thus $x \ne x_0$. If $y \ne x_0$, then since $(q,w) \sqsubseteq (q',w')$, the property holds.

   Now, assume that $y = x_0$. To conclude the proof of Property 2 we need to show that $(w'+\tau')(x) \ge (w+\tau)(x)$ or $(w'+\tau')(x) \equiv (w+\tau)(x)$. If $w(x) \equiv w'(x)$, since we have chosen τ' according to the property stated in Proposition 1(2), clearly this holds and we are done with this property.

   Otherwise, i.e., $w(x) \not\equiv w'(x)$, since $(q,w) \sqsubseteq (q',w')$, we must have that $w'(x) \ge w(x)$. Since $w(x) \not\equiv w'(x)$, we get that either (1) $\lfloor w(x) \rfloor < \lfloor w'(x) \rfloor$, or (2) $\lfloor w(x) \rfloor = \lfloor w'(x) \rfloor$, $\text{fract}(w(x)) = 0$ and $\text{fract}(w'(x)) > 0$. If the first case holds, then clearly $\lfloor w'(x) \rfloor \ge \lfloor w(x) \rfloor + 1$ also holds. Thus since $\tau \equiv \tau'$, it follows that either $w'(x)+\tau' \equiv w(x)+\tau$ or $w'(x)+\tau' \ge w(x)+\tau$, and the property holds.

   In the remaining case, i.e., when $\text{fract}(w(x)) = 0$, $\text{fract}(w'(x)) > 0$ and $\lfloor w(x) \rfloor = \lfloor w'(x) \rfloor$, we get $w'(x) > w(x) = \lfloor w'(x) \rfloor$. Suppose that $(w'+\tau')(x) < (w+\tau)(x)$ (if $(w'+\tau')(x) \ge (w+\tau)(x)$ we are done). Since $\tau \equiv \tau'$ (hence, $\lfloor \tau \rfloor = \lfloor \tau' \rfloor$) and $w(x) = \lfloor w'(x) \rfloor \in \mathbb{N}$, we get $\lfloor (w+\tau)(x) \rfloor = \lfloor w'(x) \rfloor + \lfloor \tau' \rfloor$. Hence, $\lfloor (w+\tau)(x) \rfloor \le \lfloor (w'+\tau')(x) \rfloor$. From the hypothesis $(w'+\tau')(x) < (w+\tau)(x)$, then $\lfloor (w+\tau)(x) \rfloor = \lfloor (w'+\tau')(x) \rfloor$ must hold, and therefore, $\text{fract}((w+\tau)(x)) > 0$. Suppose now by contradiction that $\text{fract}((w'+\tau')(x)) = 0$. Then, $\text{fract}(w'(x)) + \tau' = \lfloor \tau' \rfloor + 1$, but since $(w'+\tau')(x) < (w+\tau)(x)$, this implies that $\tau > \lfloor \tau' \rfloor + 1$ and thus contradicting $\tau \equiv \tau'$. Therefore, $\text{fract}((w'+\tau')(x)) > 0$ must hold. Hence, we obtain that $(w'+\tau')(x) \equiv (w+\tau)(x)$ and this concludes the proof of Property 2.
3. Property 3 can be proved similarly to Property 2.

4. If the edge $s \overset{\delta,\tau}{\twoheadrightarrow} s_1$ corresponds to a time transition, then the claim directly follows from Property 2. Now, assume that $\delta = (q, g, r, q_1)$. Then, $(v, w) \models g$, $t_1 = t$, $\tau = 0$, and $s_1 = (q_1, w[r])$. Since $(s, t) \sqsubseteq (s', t')$ (recall $s = (q, w)$ and $s' = (q, w')$), by Property 1, $((q_1, w[r]), t) \sqsubseteq ((q_1, w'[r]), t')$. Thus, it suffices to show that $((q, w'), t') \overset{\delta,0}{\twoheadrightarrow} ((q_1, w'[r]), t')$ is an extended edge of $[\![\mathcal{A}]\!]_v$, i.e. $(v, w') \models g$. Since $s' \approx_{v_{null}} s$ and $(v, w) \models g$, w' satisfies all non-parametric atomic constraints occurring in g. Now, assume that f is a parametric atomic clock constraint in g of the form $f = x - y \prec e$. Since v evaluates negative for $\mathcal{A}$, $e[v] < 0$. Also $(v, w) \models g$, hence $w(x) - w(y) < 0$. Since $s' \sqsupseteq s$, it follows that either $w'(x) - w'(y) \leq w(x) - w(y)$ or $w'(x) - w'(y) \equiv w(x) - w(y)$. Since $e[v] \in \mathbb{Z}$, we deduce that $(v, w') \models f$. Thus, we conclude that $(v, w') \models g$.
5. If the edge $s' \overset{\delta,\tau'}{\twoheadrightarrow} s'_1$ corresponds to a time transition, then the claim directly follows from Property 3. Now, assume that $\delta = (q, g, r, q_1)$. Then, $(v, w') \models g$. By Property 1, it suffices to show that $(v, w) \models g$. Since $s \approx_{v_{null}} s'$, w satisfies all non-parametric atomic constraints occurring in g. Now, assume that f is a parametric atomic clock constraint in g of the form $f = x - y \prec e$. Since v evaluates positive for $\mathcal{A}$, $e[v] > 0$. Thus, if $w(x) - w(y) \leq 0$, then $(v, w) \models f$. Now, assume that $w(x) - w(y) > 0$. Since $s \sqsubseteq s'$ and $e[v] \in \mathbb{N}$, we easily deduce that $(v, w) \models f$. Thus, we conclude that $(v, w) \models g$.

□

4 Decision problems for L/U automata

In this section, we discuss our results on L/U automata. We first prove in Sects. 4.1 and 4.2 that emptiness and universality are decidable for both lower bound and upper bound automata. Next, in Sect. 4.3 we combine these results to solve emptiness, universality, and finiteness for general L/U automata. Then, in Sect. 4.4 we extend the considered problems placing linear constraints on the parameters. Finally, in Sect. 4.5, by using the results of Sects. 4.1 and 4.2, we show that for lower bound and upper bound automata $\mathcal{A}$, the set $\Gamma(\mathcal{A})$ can be effectively represented by a linear constraint whose size is doubly exponential in the number of parameters.

For an L/U automaton $\mathcal{A}$, we will use the following constants:

- $k_{\mathcal{A}}$ denotes the number of parametric clocks of $\mathcal{A}$, i.e. the size of $X(P, \mathcal{A})$;
- $c^0_{\mathcal{A}}$ is the maximum over $\{|c| + 1 \mid$ there is a linear expression of $\mathcal{A}$ of the form $c_0 + c_1 p_1 + \cdots + c_m p_m$ and $c = c_0\}$.
- $c_{\mathcal{A}}$ is the maximum over $\{|c| + 1 \mid$ there is a linear expression of $\mathcal{A}$ of the form $c_0 + c_1 p_1 + \cdots + c_m p_m$ and $c = c_i$ for some $0 \leq i \leq m\}$.
- $N_{\mathcal{R}(\mathcal{A})}$ is the number of regions of $\mathcal{A}$ with respect to the null parameter valuation.

4.1 Emptiness and universality for lower bound automata

Given a lower bound automaton $\mathcal{A}$, recall that every linear expression of $\mathcal{A}$ is of the form $c_0 - c_1 p_1 - \cdots - c_m p_m$ with $c_i \in \mathbb{N}$ for $1 \leq i \leq m$. By decreasing the parameter values, the clock constraints of $\mathcal{A}$ are weakened. Thus, if $v \leq v'$, then each run of $\mathcal{A}_{v'}$ is also a run of $\mathcal{A}_v$. Hence, if $v \leq v'$ and $v' \in \Gamma(\mathcal{A})$, then also $v \in \Gamma(\mathcal{A})$. Therefore, we obtain the following result.

Proposition 3 *For lower bound automata $\mathcal{A}$, the set $\Gamma(\mathcal{A})$ is downward-closed.*

Moreover, note that except for a *finite* and computable set of parameter valuations depending on the specific lower bound automaton $\mathcal{A}$, a parameter valuation *evaluates negative* for $\mathcal{A}$.

Emptiness Since for lower bound automata $\mathcal{A}$, the set $\Gamma(\mathcal{A})$ is downward-closed, in order to test emptiness of $\Gamma(\mathcal{A})$ it suffices to check emptiness of the TA $\mathcal{A}_{v_{null}}$. By Theorem 1, we obtain the following result.

Theorem 2 *For lower bound automata $\mathcal{A}$, checking emptiness of $\Gamma(\mathcal{A})$ is* PSPACE*-complete and can be done in time $O(|\Delta| \cdot (2c_{\mathcal{A}}+2)^{2|X|^2})$.*

Universality For checking universality of $\Gamma(\mathcal{A})$ for a given lower bound automaton $\mathcal{A}$, we define a parameter valuation $v_{\mathcal{A}}$ (assigning "large" values to parameters), which *evaluates negative* for $\mathcal{A}$, and show that if $v_{\mathcal{A}} \in \Gamma(\mathcal{A})$ then each $v \geq v_{\mathcal{A}}$ also belongs to $\Gamma(\mathcal{A})$ (note that since $\mathcal{A}$ is a lower bound automaton, each $v \geq v_{\mathcal{A}}$ still evaluates negative for $\mathcal{A}$). Since $\Gamma(\mathcal{A})$ is downward closed, checking universality of $\Gamma(\mathcal{A})$ reduces to checking if $v_{\mathcal{A}} \in \Gamma(\mathcal{A})$, and thus, checking for non-emptiness of the TA $\mathcal{A}_{v_{\mathcal{A}}}$.

We start by giving two technical lemmata that will be used in the proof of the main theorem of this subsection. These lemmata are applicable to *unrestricted* PTA $\mathcal{A}$ under the hypothesis that the given parameter valuation v evaluates negative for $\mathcal{A}$.

Lemma 1 (First Simulation Lemma) *Let v be a parameter valuation which evaluates negative for a* PTA *$\mathcal{A}$. Let $\rho = s_0 \xrightarrow{\delta_0,\tau_0} s_1 \xrightarrow{\delta_1,\tau_1} \cdots$ be a run of $\mathcal{A}_v$ and $s'_0 \sqsupseteq s_0$. Then, there is a run of $\mathcal{A}_v$ of the form $\rho' = s'_0 \xrightarrow{\delta_0,\tau'_0} s'_1 \xrightarrow{\delta_1,\tau'_1} \cdots$ such that $s'_i \sqsupseteq s_i$ and $\mathrm{DUR}(\rho'[0,i]) \equiv \mathrm{DUR}(\rho[0,i])$ for each i.*

Proof Let $\mathrm{ext}(\rho) = (s_0,t_0) \xrightarrow{\delta_0,\tau_0} (s_1,t_1) \xrightarrow{\delta_1,\tau_1} \cdots$ be the extended run of $\mathcal{A}_v$ associated with ρ, where $t_0 = 0$ and $t_i = \mathrm{DUR}(\rho[0,i])$ for each i. Since $s'_0 \sqsupseteq s_0$, it holds that $(s'_0, 0) \sqsupseteq (s_0, 0)$. Thus, by repeatedly applying Proposition 2(4), there is an extended run of $\mathcal{A}_v$ of the form $\mathrm{ext}(\rho') = (s'_0,t'_0) \xrightarrow{\delta_0,\tau'_0} (s'_1,t'_1) \xrightarrow{\delta_1,\tau'_1} \cdots$ such that $t'_0 = 0$ and $(s'_i,t'_i) \sqsupseteq (s_i,t_i)$ for each i. Therefore, the associated run ρ' of $\mathcal{A}_v$ satisfies the lemma. □

The following technical lemma gives a sufficient condition for extending a run with another run which can be then "pumped" arbitrarily many times by repeatedly applying Lemma 1. In particular, let π be a finite run of $\mathcal{A}_v$ ending with a suffix ρ which corresponds to a cycle in the region graph of the timed automaton $\mathcal{A}_{v_{null}}$. Provided the hypothesis of the lemma are satisfied, we can then append to π a finite run ρ' such that ρ' corresponds to the same cycle of $\mathcal{A}_{v_{null}}$ as ρ, $s \sqsubseteq s'$ where s and s' are respectively the initial and final states of ρ', and the resulting run is still a run of $\mathcal{A}_v$. Note that the additional condition we require on the appended run ρ', i.e., $s \sqsubseteq s'$, is crucial to allow the repeated applications of Lemma 1. More precisely, assume that $\rho = s_0 \xrightarrow{\delta_0,\tau_0} s_1 \cdots s_{n-1} \xrightarrow{\delta_{n-1},\tau_{n-1}} s_n$ is a finite run of $\mathcal{A}_v$ (with $s_i = (q_i, w_i)$ for each $0 \leq i \leq n$) corresponding to a cycle in the region graph of $\mathcal{A}_{v_{null}}$, i.e. such that $s_0 \approx_{v_{null}} s_n$. Note that ρ is also a run of $\mathcal{A}_{v_{null}}$. Thus, by Proposition 1 it easily follows that there exists a run ρ' of $\mathcal{A}_{v_{null}}$ starting from s_n which is "similar" to ρ, i.e. such that ρ' has the form $\rho' = s'_0 \xrightarrow{\delta_0,\tau'_0} s'_1 \cdots s'_{n-1} \xrightarrow{\delta_{n-1},\tau'_{n-1}} s'_n$ with $s'_0 = s_n$ and for each i, $s'_i = (q_i, w'_i)$, $s_i \approx_{v_{null}} s'_i$, and $\mathrm{DUR}(\rho[0,i]) \equiv \mathrm{DUR}(\rho'[0,i])$. Moreover, for each clock x

and $i < n$, $w_i(x) \equiv w'_i(x)$ implies $w_{i+1}(x) \equiv w'_{i+1}(x)$. Hence, one can easily deduce that $s'_0 = s_n \sqsubseteq s'_n$.

Obviously, if no parametric clock constraint appears along ρ (hence, also along ρ'), then ρ' is also a run of $\mathcal{A}_v$, and we are done. Otherwise, in general it in not guaranteed that ρ' can be chosen in such a way it is also a run of $\mathcal{A}_v$. For example, assume that $\mathcal{A}$ has a unique location q and a unique transition given by $\delta = (q, -x \leq -l, \{x\}, q)$. Let $v(l) = 4$. Evidently, $\rho = (q,4) \xrightarrow{\delta,0} (q,0) \xrightarrow{\perp,3} (q,3)$ is a run of $\mathcal{A}_v$ such that $(q,4) \approx_{v_{null}} (q,3)$. However, there is no run of $\mathcal{A}_v$ starting from $(q,3)$ whose first step corresponds to an instantaneous transitions.

Thus, we need additional assumptions on the cycle ρ. In particular, we assume that v evaluates negative for $\mathcal{A}$ and if a parametric atomic constraint of the form $x - y \prec e$ with $y \neq x_0$ appears along ρ, then y is never reset along ρ. This ensures that ρ' is also a run of $\mathcal{A}_v$. Formally, we obtain the following result.

Lemma 2 (Pumping Lemma) *Let v be a parameter valuation which evaluates negative for a* PTA *$\mathcal{A}$. Let $\rho = s_0 \xrightarrow{\delta_0,\tau_0} s_1 \cdots s_{n-1} \xrightarrow{\delta_{n-1},\tau_{n-1}} s_n$ be a finite run of $\mathcal{A}_v$ such that $s_0 \approx_{v_{null}} s_n$ and if a parametric atomic constraint of the form $x - y \prec e$ with $y \neq x_0$ appears along ρ, then y is never reset along ρ.*

Then, there is a finite run $\rho' = s'_0 \xrightarrow{\delta_0,\tau'_0} s'_1 \cdots\ s'_{n-1} \xrightarrow{\delta_{n-1},\tau'_{n-1}} s'_n$ of $\mathcal{A}_v$ starting from $s'_0 = s_n$ such that $\mathrm{DUR}(\rho') \equiv \mathrm{DUR}(\rho)$ *and $s_n \sqsubseteq s'_n$.*

Proof

Let $(s_0,t_0) \xrightarrow{\delta_0,\tau_0} (s_1,t_1) \cdots (s_{n-1},t_{n-1}) \xrightarrow{\delta_{n-1},\tau_{n-1}} (s_n,t_n)$ be the extended run of $\mathcal{A}_v$ associated with ρ. Recall that $t_0 = 0$. Moreover, let $s_h = (q_h, w_h)$ for $0 \leq h \leq n$. We first prove the following claim.

Claim There is an extended run of $\mathcal{A}_v$ of the form $(s'_0,t'_0) \xrightarrow{\delta_0,\tau'_0} (s'_1,t'_1) \cdots (s'_{n-1},t'_{n-1}) \xrightarrow{\delta_{n-1},\tau'_{n-1}} (s'_n,t'_n)$ such that $s'_0 = s_n$, $t'_0 = 0$, and for all $0 \leq h \leq n$, $s'_h = (q_h, w'_h)$ and $(s'_h,t'_h) \approx_{v_{null}} (s_h,t_h)$. Moreover, for all $x \in X$ and $0 \leq h < n$, $w_h(x) \equiv w'_h(x)$ implies $w_{h+1}(x) \equiv w'_{h+1}(x)$.

Proof By hypothesis $s_0 \approx_{v_{null}} s_n$. Thus, setting $t'_0 = 0$ and $s'_0 = s_n$, it holds that $(s'_0,t'_0) \approx_{v_{null}} (s_0,t_0)$ holds. Now, assume the existence of an extended finite run ρ'_k of $\mathcal{A}_v$ of the form $\rho'_k = (s'_0,t'_0) \xrightarrow{\delta_0,\tau'_0} (s'_1,t'_1) \cdots (s'_{k-1},t'_{k-1}) \xrightarrow{\delta_{k-1},\tau'_{k-1}} (s'_k,t'_k)$ with $k < n$, $s'_k = (q_k, w'_k)$, and satisfying $(s'_h,t'_h) \approx_{v_{null}} (s_h,t_h)$ for all $0 \leq h \leq k$. Then, it suffices to show that there is an extended edge in $[\![\mathcal{A}]\!]_v$ of the form $(s'_k,t'_k) \xrightarrow{\delta_k,\tau'_k} (s'_{k+1},t'_{k+1})$ such that $(s'_{k+1},t'_{k+1}) \approx_{v_{null}} (s_{k+1},t_{k+1})$, $s'_{k+1} = (q_{k+1}, w'_{k+1})$, and for all $x \in X$, $w_k(x) \equiv w'_k(x)$ implies $w_{k+1}(x) \equiv w'_{k+1}(x)$.

Since $(s_k,t_k) \xrightarrow{\delta_k,\tau_k} (s_{k+1},t_{k+1})$ and $(s'_k,t'_k) \approx_{v_{null}} (s_k,t_k)$, the case of a time transition directly follows from Proposition 1(2). Now, assume that $\delta_k = (q_k, g, r, q_{k+1})$. By Proposition 1(1), it suffices to show that $(v, w'_k) \models g$. Since $s'_k \approx_{v_{null}} s_k$, w'_k satisfies all non-parametric atomic constraints occurring in g. Now, assume that f is a parametric atomic constraint of g of the form $x - y \prec e$. Since v evaluates negative for $\mathcal{A}$, $e[v] < 0$ holds. Therefore, $w_k(x) - w_k(y) < 0$, and thus, y cannot be x_0. Moreover, by hypothesis, clock y cannot be reset along ρ (hence, also along ρ'_k). Since $s'_0 = s_n$, it follows that $w'_k(y) =$

$w_k(y) + (\text{DUR}(\rho) - t_k) + t'_k$. Since $w'_k(x) \leq w_k(x) + (\text{DUR}(\rho) - t_k) + t'_k$ (x could be reset), we obtain that $w'_k(x) - w'_k(y) \leq w_k(x) - w_k(y)$, which implies $(v, w'_k) \models f$. Thus, we conclude that $(v, w'_k) \models g$. □

Let $(s'_0, t'_0) \xrightarrow{\delta_0, \tau'_0} (s'_1, t'_1) \cdots (s'_{n-1}, t'_{n-1}) \xrightarrow{\delta_{n-1}, \tau'_{n-1}} (s'_n, t'_n)$ be an extended run of $\mathcal{A}_v$ satisfying the claim above with $s'_h = (q_h, w'_h)$ for all $0 \leq h \leq n$, and $s'_0 = s_n$. Let ρ' be the associated run of $\mathcal{A}_v$. We have that $\text{DUR}(\rho') = t'_n$ and $\text{DUR}(\rho) = t_n$. By the claim above, $\text{DUR}(\rho) \equiv \text{DUR}(\rho')$. Thus, to conclude the proof of the lemma, we need to show that $s'_n \sqsupseteq s_n$. By the claim above, $s_n \approx_{v_{null}} s'_n$. Now, let $x, y \in X(P, \mathcal{A}) \cup \{x_0\}$ such that $w_n(x) - w_n(y) > 0$ (observe that $x \neq x_0$ must hold). We distinguish two cases:

- Clock x is never reset along ρ'. Then, $w'_n(x) = \text{DUR}(\rho') + w_n(x)$. Since $w'_n(y) \leq w_n(y) + \text{DUR}(\rho')$, we obtain that $w'_n(x) - w'_n(y) \geq w_n(x) - w_n(y)$.
- Clock x is reset along ρ' (hence, also along ρ). In this case, there is $0 < h \leq n$ such that x is reset on the edge $\rho[h-1, h]$ and x is not reset along $\rho[h, n]$. Assume that $y \neq x_0$ (the other case being similar). First, observe that clock y must be reset along $\rho[h, n]$. Indeed, assuming the contrary, we deduce that $w_n(x) - w_n(y) = w_h(x) - w_h(y) = -w_h(y) \leq 0$, which contradicts the hypothesis $w_n(x) - w_n(y) > 0$. Therefore, there is $h < k \leq n$, such that y is reset along $\rho[k-1, k]$ and y is not reset along $\rho[k, n]$. Thus, $w_n(x) - w_n(y) = w_k(x) - w_k(y) = w_k(x)$ and $w_h(x) = 0$. By construction, x and y are not reset also along $\rho'[k, n]$, x is reset along the edge $\rho'[h-1, h]$ and y is reset along the edge $\rho'[k-1, k]$. Thus, we obtain that $w'_n(x) - w'_n(y) = w'_k(x)$ and $w_h(x) = w'_h(x) = 0$ (hence $w'_h(x) \equiv w_h(x)$). Since $k > h$, by Claim 1, $w_k(x) \equiv w'_k(x)$. Thus, $w_n(x) - w_n(y) \equiv w'_n(x) - w'_n(y)$.

Thus, either $w'_n(x) - w'_n(y) \geq w_n(x) - w_n(y)$ or $w_n(x) - w_n(y) \equiv w'_n(x) - w'_n(y)$. Since $s_n \approx_{v_{null}} s'_n$, it follows that $s'_n \sqsupseteq s_n$. This concludes the proof of Lemma 2. □

Fix a lower bound automaton $\mathcal{A}$. Define $N_{\mathcal{A}}$ as the constant

$$N_{\mathcal{A}} = k_{\mathcal{A}}(N_{\mathcal{R}(\mathcal{A})} + 1) + c^0_{\mathcal{A}}$$

and denote by $v_{\mathcal{A}}$ the parameter valuation assigning $N_{\mathcal{A}}$ to each parameter. Note that $v_{\mathcal{A}}$ evaluates negative for $\mathcal{A}$. The choice of such a large constant is to ensure that in any accepting infinite run ρ of $\mathcal{A}_{v_{\mathcal{A}}}$ starting from the initial state, we can find subruns that can be repeatedly and consecutively simulated (by applying Lemmata 1 and 2) such that we can construct a corresponding infinite accepting run of $\mathcal{A}_v$, for every $v \geq v_{\mathcal{A}}$. More precisely, assume that a parametric atomic clock constraint f of $\mathcal{A}$ of the form $x - y \prec e$ is used along ρ. Note that $-e[v] \geq -e[v_{\mathcal{A}}]$ and $-e[v]$ represents a lower bound for the clock y. Then, $N_{\mathcal{A}}$ is sufficiently large to ensure that we can identify in ρ finite portions ρ' (of duration at least 1) corresponding to cycles in the region graph of $\mathcal{A}_{v_{null}}$ which do not use f, do not reset clock y, and precede the points in which f is used. Moreover, these cycles meet the hypothesis of Pumping Lemma, hence can be repeatedly simulated preserving the satisfaction of all constraints of $\mathcal{A}$ under valuation $v_{\mathcal{A}}$. The choice of such cycles ensures that if we repeatedly simulate them a sufficient number of times, then the resulting infinite run of $\mathcal{A}_{v_{\mathcal{A}}}$ preserves the satisfaction of the constraint $f = x - y \prec e$ also w.r.t. v. Thus, iterating the above reasoning over all the parametric constraints of $\mathcal{A}$, we obtain an infinite accepting run of $\mathcal{A}_v$ from the initial state.

Now, we prove formally that $v_{\mathcal{A}}$ is the key valuation for reducing universality to membership to $\Gamma(\mathcal{A})$ for a given lower bound automaton $\mathcal{A}$.

Theorem 3 *For a lower bound automaton $\mathcal{A}$, let v, v' be parameter valuations such that $v' \geq v \geq v_{\mathcal{A}}$. Then, $v \in \Gamma(\mathcal{A})$ implies $v' \in \Gamma(\mathcal{A})$.*

Proof Let v, v' be parameter valuations such that $v' \geq v \geq v_{\mathcal{A}}$. We can assume that each parameter appears precisely once in $\mathcal{A}$. In fact, if a parameter p appears twice, we can rename the second occurrence to p' and let $v(p') = v(p)$ and $v'(p') = v'(p)$. Note that this assumption does not affect the constant $N_{\mathcal{A}}$ which depends on the number of parameterized clocks and not on the number of parameters.

Fix a parameter p of $\mathcal{A}$. Let $f_p = z - y \prec e$ be the unique atomic constraint of $\mathcal{A}$ such that p occurs in e. We define v_p such that v_p assigns the value $v(p) + 1$ to p and $v(p')$ to all the other parameters p'. Since we can obtain v' from v by a sequence of steps, where a step corresponds to incrementing only one parameter by 1, it suffices to prove:

$$v \in \Gamma(\mathcal{A}) \quad \text{implies} \quad v_p \in \Gamma(\mathcal{A}) \tag{1}$$

Observe that since $v \geq v_{\mathcal{A}}$ and $\mathcal{A}$ is a lower bound automaton, we have that $e[v] \leq e[v_{\mathcal{A}}]$ and v evaluates negative for $\mathcal{A}$, and in particular, $e[v] < 0$ (note that in the constraint $f_p = z - y \prec e$, $-e[v] > 0$ represents a lower bound for the clock y). Therefore, if y is the zero clock x_0, f_p is unsatisfiable under valuation v and Assertion (1) trivially holds. Consider now the case $y \neq x_0$ and also assume that $z \neq x_0$ (the other case being simpler).

Let $\rho = s_0 \xrightarrow{\delta_0, \tau_0} s_1 \xrightarrow{\delta_1, \tau_1} s_2 \cdots$ be an infinite accepting run of $\mathcal{A}_v$ where $s_i = (q_i, w_i)$ for $i \geq 0$ and such that clock y is zero in s_0 (note that if s_0 is the initial state of $\mathcal{A}$, this last condition is satisfied). Then, we need to show that there is an infinite accepting run ρ' in $\mathcal{A}_{v_p}$ from s_0. In the following, we assume that the clock constraint f_p is used along ρ (in the other case, ρ is also a run of $\mathcal{A}_{v_p}$ and we are done).

In the rest of the proof, we first determine a finite portion of the run ρ of duration at least 1 that precedes the first state (along ρ) where a transition guarded by f_p is taken and such that clock y is never reset along it. Moreover, such a sample subrun is suitable for repeated simulation, i.e., it meets the hypothesis of Lemma 2. Then, simulate this finite run an arbitrary number of times by applying Lemma 2 for the first simulation and Lemma 1 for the remaining ones. We end with the simulation of the remaining suffix of the run ρ applying again Lemma 1. This ensures that the new infinite accepting run ρ' of $\mathcal{A}_v$ has a prefix π which is a run of $\mathcal{A}_{v_p}$ and corresponds to a prefix of the old run ρ containing some instantaneous transition. Moreover, either the whole run ρ' is also a run of $\mathcal{A}_{v_p}$, or clock y is zero in the last state of the prefix π. In the second case, the whole process is iterated (starting from the remaining suffix of ρ') until the resulting accepting infinite run of $\mathcal{A}_v$ starting from s_0 is also a run of $\mathcal{A}_{v_p}$.

Let M be the smallest index such that f_p is in the clock constraint δ_M of transition $s_M \xrightarrow{\delta_M, 0} s_M$. Thus, $(v, w_M) \models f_p$. Let M_y be the largest index in $[0, M]$ such that $w_{M_y}(y) = 0$ (recall that such M_y always exists since clock y is zero in s_0). Since $f_p = z - y \prec e$ and $e[v] \leq e[v_{\mathcal{A}}] < c^0_{\mathcal{A}} - N_{\mathcal{A}} < 0$, we can deduce that there is $M_z \in [0, M]$ such that $M_y < M_z$, $w_{M_z}(z) = 0$, and $\text{DUR}(\rho[M_y, M_z]) > N_{\mathcal{A}} - c^0_{\mathcal{A}}$.

Observe that in a run, each time transition can be split into an arbitrary number of time transitions. Thus, we can assume without loss of generality that for every $\tau \in \mathbb{N}$, there is $i \geq M_y$ such that $\text{DUR}(\rho[M_y, i]) = \tau$. The following claim allows us to apply Lemma 2. Its proof relies on a counting argument that uses the constant $N_{\mathcal{A}}$, and thus also gives a more concrete explanation of our choice for its value. □

Claim There is an interval $[i, j] \subseteq [M_y, M_z]$ such that $\text{DUR}(\rho[i, j]) \geq 1$, $s_i \approx_{v_{null}} s_j$, and for every clock $x \in X(P, \mathcal{A}) \setminus \{x_0\}$: if a parametric atomic constraint of the form $x' - x \prec e'$ appears along $\rho[i, j]$, then x is never reset along $\rho[i, j]$.

Proof Recall the following properties on the chosen M_y and M_z: $M_y < M_z$, $w_{M_y}(y) = 0$, $w_{M_z}(z) = 0$, clock y is *never reset* along $\rho[M_y, M_z]$, and $\text{DUR}(\rho[M_y, M_z]) > N_{\mathcal{A}} - c^0_{\mathcal{A}}$.

Let $M_y \leq K \leq M_z$ be such that $\text{DUR}(\rho[M_y, K]) = N_{\mathcal{A}} - c^0_{\mathcal{A}}$ (recall that $\text{DUR}(\rho[M_y, M_z]) > N_{\mathcal{A}} - c^0_{\mathcal{A}}$). Let $Y = \{x_1, \ldots, x_n\}$ with $n \leq k_{\mathcal{A}} - 1$ be the set of clocks in $X(P, \mathcal{A}) \setminus \{x_0\}$ which are reset along $\rho[M_y, K]$ and for $h = 1, \ldots, n$, let i_h be the smallest index in $[M_y, K]$ such that clock x_h is reset on the transition $\rho[i_h - 1, i_h]$. Assume without loss of generality that $i_1 \leq i_2 \leq \cdots \leq i_n$. We set $i_0 = M_y$ and $i_{n+1} = K + 1$. Thus, for every interval $[i_h, i_{h+1} - 1]$, $0 \leq h \leq n$, the following holds: for all $x \in X(P, \mathcal{A})$, either clock x is never reset along $\rho[i_h, i_{h+1} - 1]$ or its value is always less than $N_{\mathcal{A}} - c^0_{\mathcal{A}}$. Since for each parametric atomic constraint $f = x' - x \prec e'$ of $\mathcal{A}$, $e'[v] \leq e'[v_{\mathcal{A}}] < c^0_{\mathcal{A}} - N_{\mathcal{A}}$, we have that $(v, w) \models f$ implies $w(x) > N_{\mathcal{A}} - c^0_{\mathcal{A}}$. Hence, for every interval $[i_h, i_{h+1} - 1]$ and $x \in X(P, \mathcal{A}) \setminus \{x_0\}$, either clock x is never reset along $\rho[i_h, i_{h+1} - 1]$, or none of the parametric atomic constraints along $\rho[i_h, i_{h+1} - 1]$ is of the form $x' - x \prec e'$. Since $n + 1 \leq k_{\mathcal{A}}$, $N_{\mathcal{A}} - c^0_{\mathcal{A}} = k_{\mathcal{A}}(N_{\mathcal{R}(\mathcal{A})} + 1)$, and $\text{DUR}(\rho[i_h - 1, i_h]) = 0$ for $h = 1, \ldots, n$ (i.e., the only transition of $\rho[i_h - 1, i_h]$ is instantaneous), there is a k such that $\text{DUR}(\rho[i_k, i_{k+1} - 1]) \geq N_{\mathcal{R}(\mathcal{A})} + 1$. Recall that for each $\tau \in \mathbb{N}$ there is $i \geq M_y$ such that $\text{DUR}(\rho[M_y, i]) = \tau$, and $N_{\mathcal{R}(\mathcal{A})}$ is the number of equivalence classes induced by $\approx_{v_{null}}$. Hence, there are indexes $i, j \in [i_k, i_{k+1} - 1]$ such that $\text{DUR}(\rho[i, j]) \geq 1$ and $s_i \approx_{v_{null}} s_j$. Therefore, the claim holds. □

Let $[i, j] \subseteq [M_y, M_z]$ be an interval satisfying the above claim. We can apply Lemma 2 to $\rho[i, j]$ obtaining a finite run ρ_1 of $\mathcal{A}_v$ starting from s_j and leading to $s'_j \sqsupseteq s_j$. Thus we can repeatedly apply Lemma 1, to append an arbitrary number d of simulations of ρ_1 and then simulate the remaining part of ρ. Let $\rho' = \rho[0, j]\,\rho_1\rho_2..\rho_d\,\rho'_{M_z}\,\rho''$ be the obtained run of $\mathcal{A}_v$, where for $h = 2, \ldots, d$, runs ρ_h are the simulations of ρ_1, ρ'_{M_z} is the simulation of $\rho[j, M_z]$, and ρ'' is the simulation of the remaining suffix of ρ. Since f_p is not used along $\rho[0, M]$, by Lemmas 1 and 2 ρ' is an accepting infinite run of $\mathcal{A}_v$ and the clock constraint f_p never appears along $\eta = \rho[0, j]\,\rho_1\rho_2..\rho_d\,\rho'_{M_z}$, hence η is also a finite run of $\mathcal{A}_{v_p}$. Moreover, $\text{DUR}(\rho_h) \equiv \text{DUR}(\rho[i, j])$ for $h = 1, \ldots, d$, and y is not reset in $\rho_1\rho_2..\rho_d\,\rho'_{M_z}$ (recall that y is not reset along $\rho[M_y, M_z]$).

Let $s = (q, w)$ be the last state of ρ'_{M_z}. Since $s \sqsupseteq s_{M_z}$ and $w_{M_z}(z) = 0$, we have $w(z) = 0$. Since y is not reset in $\rho_1\rho_2..\rho_d\,\rho'_{M_z}$ and $\text{DUR}(\rho[i, j]) \geq 1$, by carefully choosing d, we get that $(v_p, w) \models f_p$. Thus, if clock y is never reset along ρ'', then ρ'' is also a run in $\mathcal{A}_{v_p}$, hence ρ' is an infinite accepting run of $\mathcal{A}_{v_p}$. Otherwise, there is a non empty prefix π of ρ'' (containing some instantaneous transition) such that $\rho[0, j]\,\rho_1\rho_2..\rho_d\,\rho'_{M_z}\,\pi$ is a run of $\mathcal{A}_{v_p}$ and the remaining suffix of ρ'' starts at a state in which clock y is zero. By iterating the above reasoning starting from this suffix of ρ'' we get an accepting infinite run of $\mathcal{A}_{v_p}$ starting from s_0, and the theorem is proved.

Since $\Gamma(\mathcal{A})$ is downward-closed, by the above theorem checking universality reduces to check non-emptiness of the TA $\mathcal{A}_{v_{\mathcal{A}}}$. Since the largest constant in $\mathcal{A}_{v_{\mathcal{A}}}$ is bounded by $|P| \cdot N_{\mathcal{A}} \cdot c_{\mathcal{A}}$ and $N_{\mathcal{A}} = O(|Q| \cdot k_{\mathcal{A}} \cdot (2c^0_{\mathcal{A}} + 2)^{2|X|^2})$, by Theorem 1 we obtain the following.

Theorem 4 *For lower bound automata $\mathcal{A}$, checking for the universality of $\Gamma(\mathcal{A})$ is* PSPACE-*complete and can be done in time exponential in $|X|^4$ and in the size of the encoding of $c_{\mathcal{A}}$, and polynomial in the number of parameters and locations of $\mathcal{A}$.*

4.2 Emptiness and universality for upper bound automata

The arguments used to show the results for upper bound automata are similar to those used for lower bound automata. Recall that every linear expression of a upper bound automaton $\mathcal{A}$ is of the form $c_0 + c_1 p_1 + \cdots + c_m p_m$ with $c_i \in \mathbb{N}$ for each $1 \leq i \leq m$. Thus, by increasing the parameter values, the clock constraints of $\mathcal{A}$ are weakened, hence the set $\Gamma(\mathcal{A})$ is upward-closed. An immediate consequence of this property is that testing universality of $\Gamma(\mathcal{A})$ reduces to checking non-emptiness of the TA $\mathcal{A}_{v_{null}}$. Therefore, we obtain the following result.

Theorem 5 *For upper bound automata $\mathcal{A}$, checking universality of $\Gamma(\mathcal{A})$ is* PSPACE-*complete and can be done in time $O(|\Delta| \cdot (2c_{\mathcal{A}} + 2)^{2|X|^2})$.*

For checking emptiness of $\Gamma(\mathcal{A})$, we establish a version of Theorem 3 for upper bound automata: we define a parameter valuation $v_{\mathcal{A}}$, which *evaluates positive* for $\mathcal{A}$, and show that if $v \geq v_{\mathcal{A}}$ and $v \in \Gamma(\mathcal{A})$, then $v_{\mathcal{A}}$ also belongs to $\Gamma(\mathcal{A})$ (note that since $\mathcal{A}$ is an upper bound automaton, each $v \geq v_{\mathcal{A}}$ still evaluates positive for $\mathcal{A}$). The main argument of the proof is the selection of finite portions of the considered infinite run of $\mathcal{A}_v$ that is possible to "delete" from it. Since $\Gamma(\mathcal{A})$ is upward closed, checking non-emptiness of $\Gamma(\mathcal{A})$ reduces to checking if $v_{\mathcal{A}} \in \Gamma(\mathcal{A})$, and thus, checking for non-emptiness of the TA $\mathcal{A}_{v_{\mathcal{A}}}$.

We start by giving two technical lemmata that are applicable to *unrestricted* PTA $\mathcal{A}$ under the hypothesis that the given parameter valuation v evaluates positive for $\mathcal{A}$. Lemma 3 directly follows from Proposition 2(5).

Lemma 3 (Second Simulation Lemma) *Let v be a parameter valuation which* evaluates positive *for a* PTA $\mathcal{A}$. *Let $\rho = s_0 \xrightarrow{\delta_0, \tau_0} s_1 \xrightarrow{\delta_1, \tau_1} \cdots$ be a run of $\mathcal{A}_v$ and $s'_0 \sqsubseteq s_0$. Then, there is a run of $\mathcal{A}_v$ of the form $\rho' = s'_0 \xrightarrow{\delta_0, \tau'_0} s'_1 \xrightarrow{\delta_1, \tau'_1} \cdots$ such that $s'_i \sqsubseteq s_i$ and* $\mathrm{DUR}(\rho'[0, i]) \equiv \mathrm{DUR}(\rho[0, i])$ *for each i.*

The following technical lemma allow us to replace a finite run $\pi\rho$ of $\mathcal{A}_v$ starting from s_0 and leading to s such that π corresponds to a cycle in the region graph of $\mathcal{A}_{v_{null}}$ (of duration at least 1) with a new run ρ' "similar" to ρ, starting in s_0, leading to a state s' satisfying $s' \sqsubseteq s$, and such that $\mathrm{DUR}(\rho') \equiv \mathrm{DUR}(\rho)$. Note that if $\pi\rho$ is a subrun of an infinite run $\rho_0\pi\rho\rho_1$, then applying the Simulation Lemma we can construct a new infinite run $\rho_0\rho'\rho'_1$, where ρ'_1 is a simulation of ρ_1. Intuitively, the lemma allows us to "delete" the cycle π from the infinite run $\rho_0\pi\rho\rho_1$.

Lemma 4 (Cutting Lemma) *Let v be a parameter valuation which* evaluates positive *for a* PTA $\mathcal{A}$. *Let $\pi\rho$ be a run of $\mathcal{A}_v$ from $s'_0 = (q_0, w'_0)$ with $\rho = s_0 \xrightarrow{\delta_0, \tau_0} s_1 \cdots s_{n-1} \xrightarrow{\delta_{n-1}, \tau_{n-1}} s_n$ such that $s'_0 \approx_{v_{null}} s_0$,* $\mathrm{DUR}(\pi) \geq 1$, *and for each non-constant linear expression e of $\mathcal{A}$ and parametric clock $x \neq x_0$: if x is reset along π, then x is reset along ρ and $w'_0(x) +$* $\mathrm{DUR}(\pi\rho) < e[v]$.

Then, there is a finite run $\rho' = s'_0 \xrightarrow{\delta_0, \tau'_0} s'_1 \cdots s'_{n-1} \xrightarrow{\delta_{n-1}, \tau'_{n-1}} s'_n$ of $\mathcal{A}_v$ such that $\mathrm{DUR}(\rho') \equiv \mathrm{DUR}(\rho)$, *and $s'_n \sqsubseteq s_n$.*

The proof of Lemma 4 is similar to that of Lemma 2, and is given in full detail in Appendix A.1. Here, we just provide a sketched proof. Let $\pi\rho$ be a run of $\mathcal{A}_v$ from $s'_0 = (q_0, w'_0)$

with $\rho = s_0 \xrightarrow{\delta_0,\tau_0} s_1 \cdots s_{n-1} \xrightarrow{\delta_{n-1},\tau_{n-1}} s_n$ such that $s'_0 \approx_{v_{null}} s_0$ ($s_i = (q_i, w_i)$ for each $0 \leq i \leq n$). If no parametric clock constraint appears along ρ, then ρ is also a run of $\mathcal{A}_{v_{null}}$. Thus, by Proposition 1 it easily follows that there exists a run ρ' of $\mathcal{A}_{v_{null}}$ starting from s'_0 which is "similar" to ρ, i.e. such that ρ' has the form $\rho' = s'_0 \xrightarrow{\delta_0,\tau'_0} s'_1 \cdots s'_{n-1} \xrightarrow{\delta_{n-1},\tau'_{n-1}} s'_n$, where for each i, $s'_i = (q_i, w'_i)$, $s_i \approx_{v_{null}} s'_i$, and $\mathrm{DUR}(\rho[0,i]) \equiv \mathrm{DUR}(\rho'[0,i])$. Moreover, for each clock x and $i < n$, $w_i(x) \equiv w'_i(x)$ implies $w_{i+1}(x) \equiv w'_{i+1}(x)$. Note that ρ' is also a run of $\mathcal{A}_v$. Without more assumptions on the run ρ, one cannot conclude that $s'_n \sqsubseteq s_n$. This is because we also assume that $\mathrm{DUR}(\pi) \geq 1$ and for each parametric clock $x \neq x_0$: if x is reset along π, then x is reset along ρ. Note that this ensures that if x is not reset along ρ, then x is not reset also along ρ' and $w_0(x) = w'_0(x) + \mathrm{DUR}(\pi) \geq w'_0(x) + 1$. One can easily deduce that under this assumption $s'_n \sqsubseteq s_n$ holds. Now, suppose that some parametric clock constraint appears along ρ. Without further assumptions on the run ρ, in this case the existence of ρ' is not guaranteed. This is because we need to assume that v evaluates negative for $\mathcal{A}$ and for each parametric clock $x \neq x_0$ and linear expression e of $\mathcal{A}$: if x is reset along π, then $w'_0(x) + \mathrm{DUR}(\pi\rho) < e[v]$. For more details, see Appendix A.1.

Now, we establish a version of Theorem 3 for upper bound automata $\mathcal{A}$. Here, we use a slightly larger constant

$$N_{\mathcal{A}} = 8k_{\mathcal{A}}(N_{\mathcal{R}(\mathcal{A})} + 1) + c^0_{\mathcal{A}}$$

The definition of such constant is again motivated by counting arguments as in the case of lower bound automata. Define $v_{\mathcal{A}}$ as the valuation assigning $N_{\mathcal{A}}$ to each parameter.

Theorem 6 *For an upper bound automaton $\mathcal{A}$, let v, v' be parameter valuations such that $v \geq v' \geq v_{\mathcal{A}}$. Then, $v \in \Gamma(\mathcal{A})$ implies $v' \in \Gamma(\mathcal{A})$.*

Proof Note that the constant $N_{\mathcal{A}}$ does not depends on the number of parameters and for each linear expression $e = c_0 + c_1 p_1 + \cdots + c_m p_m$ of $\mathcal{A}$, $N_{\mathcal{A}}$ does not depends on the non-negative coefficients $c_1, \ldots, c_m$ of parameters $p_1, \ldots, p_m$. Thus, we can assume that each parameter appears precisely once in $\mathcal{A}$ and each coefficient $c_i > 0$ (with $1 \leq i \leq m$) in a linear expression $e = c_0 + c_1 p_1 + \cdots + c_m p_m$ of $\mathcal{A}$ is exactly 1 (if $c_i > 1$, we introduce fresh parameters $p'_1, \ldots, p'_h$ with $h = c_i - 1$, replace $c_i p_i$ with the expression $p_i + p'_1 + \cdots + p'_h$, and let $v(p'_j) = v(p_i)$ and $v'(p'_j) = v'(p_i)$ for each $1 \leq j \leq h$).

Fix a parameter p of $\mathcal{A}$. Let $f_p = y - z \prec e$ be the unique atomic constraint of $\mathcal{A}$ such that p occurs in f_p. Define v_p such that v_p assigns the value $v(p) - 1$ to p and $v(p')$ to all the other parameters p'. Note that by the above assumption, $e[v] - e[v_p] = 1$. Since we can obtain v' from v by a sequence of steps, where a step corresponds to decrementing one parameter by one, in order to prove Theorem 6, it suffices to prove that

$$v \in \Gamma(\mathcal{A}) \quad \text{implies} \quad v_p \in \Gamma(\mathcal{A}) \tag{2}$$

Observe that since $v_p \geq v_{\mathcal{A}}$, v_p evaluates positive for $\mathcal{A}$, hence $e[v_p] > 0$. It follows that if $y = x_0$, then the constraint f_p is fulfilled by every clock valuation under the parameter valuation v_p. Thus, in this case Assertion (2) trivially holds. Therefore, in the following we assume that $y \neq x_0$. We also assume that $z \neq x_0$ (the other case being simpler).

Let $\rho = s_0 \xrightarrow{\delta_0,\tau_0} s_1 \xrightarrow{\delta_1,\tau_1} s_2 \cdots$ be an infinite accepting run of $\mathcal{A}_v$ where $s_i = (q_i, w_i)$ for $i \geq 0$ and such that clock y is zero in s_0 (note that if s_0 is the initial state of $\mathcal{A}$, this last condition is satisfied). Then, we need to show that there is an infinite accepting run ρ' in $\mathcal{A}_{v_p}$ from s_0. Assume that ρ is not a run of $\mathcal{A}_{v_p}$ (otherwise, we are done).

In the rest of the proof, we first determine a suffix $\eta\rho_1$ of the run ρ of $\mathcal{A}_v$, where the portion η precedes the first state (along ρ) whose clock evaluation does not satisfy f_p under v_p and such that clock y is never reset along η. Moreover, such a sample subrun η meets the hypothesis of the Cutting Lemma. Thus, by applying Lemma 3 we can replace the suffix $\eta\rho_1$ in ρ with a new run $\eta'\rho'_1$ of $\mathcal{A}_v$ (where ρ'_1 is a simulation of ρ_1) such that clock y is never reset along η' and $\mathrm{DUR}(\eta) - \mathrm{DUR}(\eta') \geq 2$. Finally, we show that the resulting infinite accepting run ρ' of $\mathcal{A}_v$ satisfies the following: either ρ' is also a run of $\mathcal{A}_{v_p}$, or there is a prefix of ρ' (containing some instantaneous transition) which is a run of $\mathcal{A}_{v_p}$ and such that clock y is zero in its last state. In the second case, the whole process is iterated (starting from the remaining suffix of ρ') until the resulting accepting infinite run starting from s_0 is also a run of $\mathcal{A}_{v_p}$. Now, we give the technical details.

Let M be the smallest index such that $(v_p, w_M) \not\models f_p$. Note that $\rho[0, M]$ is also a run of $\mathcal{A}_{v_p}$. Let M_y be the largest index in $[0, M]$ such that $w_{M_y}(y) = 0$ (recall that clock y is zero in s_0). Since $f_p = y - z \prec e$ and $e[v_p] \geq e[v_{\mathcal{A}}] > N_{\mathcal{A}} - c^0_{\mathcal{A}} > 0$, by simple arguments, we deduce the following:

- there is $M_z \in [0, M]$ such that $M_y < M_z$, $w_{M_y}(y) = 0$, $w_{M_z}(z) = 0$, clock y is *never reset* along $\rho[M_y, M_z]$, and $\mathrm{DUR}(\rho[M_y, M_z]) > N_{\mathcal{A}} - c^0_{\mathcal{A}}$.

Since each time transition can be split into an arbitrary number of time transitions, w.l.o.g. we can assume that for every $\tau \in \mathbb{N} \cap [0, N_{\mathcal{A}} - c^0_{\mathcal{A}}]$, there is $M_y \leq i \leq M_z$ such that $\mathrm{DUR}(\rho[i, M_z]) = \tau$. The following claim allows us to apply Lemma 4 for the parameter valuation v_p. Its proof relies on a counting argument that uses the constant $N_{\mathcal{A}}$, and thus also gives a more concrete explanation of our choice for its value. □

Claim 1 There is an interval $[i, j] \subseteq [M_y, M_z]$ such that $\mathrm{DUR}(\rho[i, j]) \geq 2$, $s_i \approx_{v_{null}} s_j$, $\mathrm{DUR}(\rho[j, M_z]) \in \mathbb{N} \setminus \{0\}$, and for each non-constant linear expression e' of $\mathcal{A}$ and parametric clock $x \neq x_0$: if x is reset along $\rho[i, j]$, then x is reset along $\rho[j, M_z]$ and $w_i(x) + \mathrm{DUR}(\rho[i, M_z]) < e'[v_p]$. Also, $\rho[M_y, M_z]$ is accepting iff $\rho[M_y, i]$ or $\rho[j, M_z]$ is accepting.

Proof Let $M_y \leq K \leq M_z$ such that $\rho[K, M_z] = N_{\mathcal{A}} - c^0_{\mathcal{A}}$. Note that for every non-constant linear expression e' of $\mathcal{A}$, $e'[v_p] \geq e'[v_{\mathcal{A}}] > N_{\mathcal{A}} - c^0_{\mathcal{A}}$. Let $Y = \{x_1, \ldots, x_n\}$ with $n \leq k_{\mathcal{A}} - 1$ be the set of clocks in $X(P, \mathcal{A}) \setminus \{x_0\}$ which are reset along $\rho[K, M_z]$ and for every $1 \leq h \leq n$, let i_h (resp., j_h) be the smallest (resp., the greatest) index i in $[K, M_z]$ such that clock x_h is reset on the transition $\rho[i-1, i]$. Let $i_0 = K$ and $i_{n+1} = M_z + 1$, and let $k_0 \leq k_1 \leq \cdots \leq k_{2n+1}$ be an ordering of the indexes $i_0, \ldots, i_{n+1}, j_1, \ldots, j_n$. Thus, for every interval $[k_h, k_{h+1} - 1]$ (for $0 \leq h \leq 2n$), non-constant linear expression e' of $\mathcal{A}$, and $x \in X(P, \mathcal{A}) \setminus \{x_0\}$, the following holds: if clock x is reset along $\rho[k_h, k_{h+1} - 1]$, then it is reset along $\rho[k_{h+1} - 1, M_z]$ and $w_{k_h}(x) + \mathrm{DUR}(\rho[k_h, M_z]) \leq N_{\mathcal{A}} - c^0_{\mathcal{A}} < e'[v_p]$. Since $2n + 1 \leq 2k_{\mathcal{A}}$, $N_{\mathcal{A}} - c^0_{\mathcal{A}} = 8k_{\mathcal{A}}(N_{\mathcal{R}(\mathcal{A})} + 1)$, and $\mathrm{DUR}(\rho[i_h - 1, i_h]) = 0$ and $\mathrm{DUR}(\rho[j_h - 1, j_h]) = 0$ for all $1 \leq h \leq n$, there is $0 \leq r \leq 2n$ such that $\mathrm{DUR}(\rho[k_r, k_{r+1} - 1]) \geq 4(N_{\mathcal{R}(\mathcal{A})} + 1)$. Recall that for each $\tau \in \mathbb{N} \cap [0, N_{\mathcal{A}} - c^0_{\mathcal{A}}]$, there is $M_y \leq i \leq M_z$ such that $\mathrm{DUR}(\rho[i, M_z]) = \tau$, and $N_{\mathcal{R}(\mathcal{A})}$ is the number of equivalences classes induced by $\approx_{v_{null}}$. Since $\mathrm{DUR}(\rho[k_r, k_{r+1} - 1]) \geq 4(N_{\mathcal{R}(\mathcal{A})} + 1)$, it follows that there is an interval $[i, j] \subseteq [k_r, k_{r+1} - 1]$ with $s_i \approx_{v_{null}} s_j$ satisfying the claim.

Let $[i, j] \subseteq [M_y, M_z]$ with $\mathrm{DUR}(\rho[i, j]) \geq 2$ be an interval satisfying the claim above. We can apply Lemma 4 to $\rho[i, j]\rho[j, M_z]$ (with $\mathrm{DUR}(\rho[j, M_z]) \in \mathbb{N} \setminus \{0\}$) obtaining a finite run ρ_1 of $\mathcal{A}_{v_p}$ starting from s_i and leading to $s'_{M_z} = (q_{M_z}, w'_{M_z}) \sqsubseteq s_{M_z}$ such that clock y is not reset along ρ_1 (recall that y is not reset along $\rho[M_y, M_z]$), $\mathrm{DUR}(\rho_1) = \mathrm{DUR}(\rho[j, M_z])$, and ρ_1 is accepting iff $\rho[j, M_z]$ is accepting. It follows that

$$w_{M_z}(y) - w'_{M_z}(y) = \mathrm{DUR}(\rho[i, j]) \geq 2 \tag{3}$$

Note that since $v_p \leq v$ and $\mathcal{A}$ is an upper bound automaton, $\rho[0, i]\rho_1$ is both a finite run of $\mathcal{A}_{v_p}$ and $\mathcal{A}_v$ and by the claim above, it is accepting iff $\rho[0, M_z]$ is accepting.

Let ρ_2 be the suffix of ρ starting from s_{M_z} and let ρ'_2 be a simulation of ρ_2 in $\mathcal{A}_v$ starting from s'_{M_z} (whose existence is guaranteed by Lemma 3). Moreover, let $\rho' = \rho[0, i]\rho_1\rho'_2$ be the resulting accepting infinite run of $\mathcal{A}_v$. Now, we show the following.

Claim 2 For each prefix π of ρ'_2, if y is not reset along π, then π is a run of $\mathcal{A}_{v_p}$.

Proof Assume the contrary and derive a contradiction. Since $w_{M_z}(z) = 0$ and $s'_{M_z} \sqsubseteq s_{M_z}$, it holds that $w'_{M_z}(z) = 0$. Thus, it easily follows that there must be a prefix π' of ρ'_2 such that denoted by π the corresponding prefix of ρ_2, by $s' = (q', w')$ the last state of π', and by $s = (q', w)$ the last state of π, the following holds: clock z is zero in s and s', y is not reset along π and π', $(v_p, w') \not\models f_p$, and $(v, w) \models f_p$. Hence, $w(y) - w'(y) < e[v] - e[v_p]$ and $w(y) - w'(y) = w_{M_z}(y) - w'_{M_z}(y) + \mathrm{DUR}(\pi) - \mathrm{DUR}(\pi')$. Since $e[v_p] = e[v] - 1$ and $\mathrm{DUR}(\pi) \equiv \mathrm{DUR}(\pi')$ (recall that ρ'_2 is a simulation of ρ_2), it follows that $w_{M_z}(y) - w'_{M_z}(y) < 2$, which contradicts the fact that $w_{M_z}(y) - w'_{M_z}(y) \geq 2$. Thus, the claim holds. □

Thus, by Claim 2, if y is never reset along ρ'_2, then $\rho' = \rho[0, i]\rho_1\rho'_2$ is the desired accepting infinite run of $\mathcal{A}_{v_p}$. Otherwise, there is a non-empty prefix π of ρ'_2 (containing some instantaneous transition) such that $\rho[i, j]\rho_1\pi$ is a finite run of $\mathcal{A}_{v_p}$ and the remaining suffix of ρ'_2 starts at a state in which clock y is zero. By iterating the above reasoning starting from this suffix of ρ'_2 we obtain an infinite accepting run of $\mathcal{A}_{v_p}$ starting from s_0, and the theorem is proved. □

Since $\Gamma(\mathcal{A})$ is upward-closed, Theorem 6 implies that $\Gamma(\mathcal{A})$ is not empty iff $v_{\mathcal{A}} \in \Gamma(\mathcal{A})$. Thus, checking emptiness of $\Gamma(\mathcal{A})$ reduces to checking emptiness of the timed automaton $\mathcal{A}_{v_{\mathcal{A}}}$. Since the largest constant in $\mathcal{A}_{v_{\mathcal{A}}}$ is bounded by $|P| \cdot N_{\mathcal{A}} \cdot c_{\mathcal{A}}$ and $N_{\mathcal{A}} = O(|Q| \cdot k_{\mathcal{A}} \cdot (2c^0_{\mathcal{A}} + 2)^{2|X|^2})$, by Theorem 1 we obtain the following result.

Theorem 7 *For upper bound automata $\mathcal{A}$, checking for emptiness of $\Gamma(\mathcal{A})$ is* PSPACE-*complete and can be done in time exponential in $|X|^4$ and in the size of the encoding of $c_{\mathcal{A}}$, and polynomial in the number of parameters and locations of $\mathcal{A}$.*

4.3 Emptiness, universality, and finiteness for L/U automata

Given an L/U automaton $\mathcal{A}$, if we instantiate the lower bound parameters of $\mathcal{A}$, we get an upper bound automaton and, similarly, if we instantiate the upper bound parameters of $\mathcal{A}$, we get a lower bound automaton. Furthermore, monotonicity properties continue to hold: if $v \in \Gamma(\mathcal{A})$ and v' is such that $v'(p) \leq v(p)$ for each lower bound parameter p and $v'(p) \geq v(p)$ for each upper bound parameter p, then $v' \in \Gamma(\mathcal{A})$. By Theorems 3 and 6, it follows that

- To check for non-emptiness of $\Gamma(\mathcal{A})$, it suffices to check for non-emptiness of the timed automaton resulting from setting all the lower bound parameters to 0 and all the upper bound parameters to $N_{\mathcal{A}} = 8k_{\mathcal{A}}(N_{\mathcal{R}(\mathcal{A})} + 1) + c^0_{\mathcal{A}}$ (the constant defined in Sect. 4.2).

- To check for universality of $\Gamma(\mathcal{A})$, it suffices to check for non-emptiness of the timed automaton resulting from setting all the upper bound parameters to 0 and all the lower bound parameters to $N_{\mathcal{A}} = k_{\mathcal{A}}(N_{\mathcal{R}(\mathcal{A})} + 1) + c^0_{\mathcal{A}}$ (the constant defined in Sect. 4.1).

Now, let us consider the finiteness problem. We distinguish two cases:

- the set of upper bound parameters of $\mathcal{A}$ is *not* empty: for the above observations on the monotonicity properties of $\Gamma(\mathcal{A})$, we get that $\Gamma(\mathcal{A})$ is infinite if and only if $\Gamma(\mathcal{A})$ is not empty;
- the set of upper bound parameters of $\mathcal{A}$ is empty, i.e. $\mathcal{A}$ is a lower bound automaton: let $N_{\mathcal{A}} = k_{\mathcal{A}}(N_{\mathcal{R}(\mathcal{A})}+1)+c^0_{\mathcal{A}}$ (the constant defined in Sect. 4.1) and for each parameter p, let $\mathcal{A}^p$ be the lower bound automaton resulting from setting each parameter $p' \neq p$ to 0. Note that $\mathcal{A}^p$ has a unique parameter and $N_{\mathcal{A}^p} \leq N_{\mathcal{A}}$. Thus, by Theorem 3, $\Gamma(\mathcal{A}^p)$ is infinite *if and only if* the parameter valuation assigning $N_{\mathcal{A}}$ to p is in $\Gamma(\mathcal{A}^p)$, and hence, *if and only if* $v_{\mathcal{A}^p} \in \Gamma(\mathcal{A})$, where $v_{\mathcal{A}^p}(p) = N_{\mathcal{A}}$ and $v_{\mathcal{A}^p}(p') = 0$ for each $p' \neq p$. Note the if $\Gamma(\mathcal{A}^p)$ is infinite, then $\Gamma(\mathcal{A})$ is infinite. Moreover, since $\Gamma(\mathcal{A})$ is downward-closed, if $\Gamma(\mathcal{A})$ is infinite, then there must be a parameter p of $\mathcal{A}$ such that $v_{\mathcal{A}^p} \in \Gamma(\mathcal{A})$. Therefore, we conclude that $\Gamma(\mathcal{A})$ is infinite if and only if there is a parameter p of $\mathcal{A}$ such that $v_{\mathcal{A}^p} \in \Gamma(\mathcal{A})$.

Thus by Theorem 1 and the above arguments, we obtain the following result.

Theorem 8 *For L/U automata $\mathcal{A}$, checking for the emptiness (resp. universality, resp. finiteness) of $\Gamma(\mathcal{A})$ is* PSPACE*-complete and can be done in time exponential in $|X|^4$ and the size of the encoding of $c_{\mathcal{A}}$, and polynomial in the number of parameters and locations of $\mathcal{A}$.*

4.4 Linearly constrained parameters

In this subsection we consider the constrained versions of the emptiness and universality problems on L/U automata, where the constraint is represented by a system of linear equations and inequalities over parameters. We show that constrained emptiness and constrained universality are undecidable for L/U automata (the main reason being that a linear constraint can be used to force a lower bound parameter to be equal to an upper bound parameter, thus removing the restriction that has been placed on L/U automata). Decidability can be regained if we keep lower bound and upper bound parameters separated also in the linear constraint. In this case our approach relies on a bound for the set of *minimal* solutions of a linear constraint, given by Pottier [17], and our results on unconstrained emptiness and universality.

Linear constraints An *atomic linear constraint* is an expression of the form $e \sim 0$, where e is a linear expression and $\sim \in \{<, =\}$. When $\sim$ is $=$ (resp., $<$), the constraint is called an *equation* (resp., *inequality*). A *linear constraint* C is a boolean combination of atomic linear constraints (using $\vee, \wedge, \neg$). A parameter valuation v is a *solution* of C if the boolean expression obtained from C by replacing each inequality/equation $e \sim 0$ with the truth value of $e[v] \sim 0$, evaluates to true. With $\mathrm{Sol}(C)$ we denote the set of C solutions. We assume that a linear constraint C is always written as a disjunction $C_1 \vee \cdots \vee C_N$ where each C_i is a conjunction of atomic linear constraints. When $N = 1$, C is called *conjunctive linear constraint*.

A conjunctive linear constraint C of k atomic constraints can be written as $\mathbf{Bp} \sim \mathbf{b}$, where $\sim \in \{<, =\}^k$, $\mathbf{B}$ and $\mathbf{b}$ are respectively $(k \times m)$ and $(k \times 1)$ matrices of integers, and $\mathbf{p}$ is the

$(m \times 1)$ matrix of parameters (where $m = |P|$ is the number of parameters). We denote by C^{hom} the conjunctive linear constraint corresponding to $\mathbf{Bp} \sim \mathbf{0}$, where $\mathbf{0}$ is the $(k \times 1)$ null matrix. Also, we denote with $||\mathbf{B}||$ the value $\max_i\{\sum_j |b_{ij}|\}$ (b_{ij} is the element at row i and column j of $\mathbf{B}$) and with $||\mathbf{b}||$ the maximum of the absolute value of all elements in $\mathbf{b}$.

For two parameter valuations v_1 and v_2 and a constant $c \in \mathbb{N}$, the valuations $v_1 + v_2$ and $c \cdot v_1$ are defined in the standard way.

Fix a linear constraint $C = C_1 \vee C_2 \vee \cdots \vee C_N$, where each $C_i = \mathbf{B}^i\mathbf{p} \sim_i \mathbf{b}^i$ is a conjunctive linear constraint. With B (resp., b) we denote the maximum of $||\mathbf{B}^i||$ (resp., $\|\mathbf{b}^i\|$) for $i = 1, \ldots, N$. Let κ (resp., n_e) be the maximum number of atomic linear constraints (resp., equalities) over $C_1, C_2, \ldots, C_N$. Let Γ_P be the set of parameter valuations v such that for all $p \in P$, $v(p) \leq (B + b + 2)^{|P|+\kappa+n_e}$. By [17], for each conjunctive linear constraint C_i we have the following characterization of $\mathrm{Sol}(C_i)$:

$$\mathrm{Sol}(C_i) = \{v_0 + c_1 v_1^i + \cdots + c_{n_i} v_{n_i}^i \mid v_0 \in S^i \text{ and } c_1, \ldots, c_{n_i} \in \mathbb{N}\}, \tag{4}$$

where S^i and $\{v_1^i, \ldots, v_{n_i}^i\}$ are respectively the finite sets of all the solutions of C_i and all the solutions of C_i^{hom} in the *Pottier* bounded set Γ_P [17].

Constrained decision problems for L/U automata We consider the following decision problems: given an L/U automaton $\mathcal{A}$ and a linear constraint C over the $\mathcal{A}$ parameters,

- *Constrained emptiness*: is the set $\Gamma(\mathcal{A}) \cap \mathrm{Sol}(C)$ empty?
- *Constrained universality*: does $\Gamma(\mathcal{A}) \supseteq \mathrm{Sol}(C)$ hold?

In the rest of this section, we first show that constrained emptiness and universality are decidable for both lower bound automata and upper bound automata, then we conclude showing undecidability for general L/U automata.

Decidability of constrained decision problems for lower bound automata Let $\mathcal{A}$ be a lower bound automaton. We consider first the constrained emptiness problem. From the above characterization, we have that if $v \in \mathrm{Sol}(C_i)$ then $v \geq v'$ for some $v' \in S^i$. Since $\Gamma(\mathcal{A})$ is downward-closed, we get: $\Gamma(\mathcal{A}) \cap \mathrm{Sol}(C_i) \neq \emptyset$ iff $\Gamma(\mathcal{A}) \cap S^i \neq \emptyset$.

Consider now the constrained universality problem. For every $v \in S^i$, let Λ_v^i be the set of parameter valuations of the form $v + c_1 v_1^i + \cdots + c_{n_i} v_{n_i}^i$ with $c_j \in \mathbb{N}$ for all $1 \leq j \leq n_i$. Thus, $\mathrm{Sol}(C_i) = \bigcup_{v \in S^i} \Lambda_v^i$ and the problem is reduced to check that $\Gamma(\mathcal{A}) \supseteq \Lambda_v^i$ for all $v \in S^i$. Let $P_i = \{p \in P \mid v_j^i(p) = 0 \text{ for all } 1 \leq j \leq n_i\}$. Note that for all $v' \in \Lambda_v^i$ and $p \in P_i$, $v'(p) = v(p)$. Denoting by $\mathcal{A}(v)$ the lower bound automata obtained from $\mathcal{A}$ by replacing each parameter $p \in P_i$ with $v(p)$, and by $\widetilde{v}$ a minimal parameter valuation in Λ_v^i such that for all $p \in P \setminus P_i$, $\widetilde{v}(p) \geq N_{\mathcal{A}(v)}$ (where $N_{\mathcal{A}(v)}$ is defined as in Sect. 4.1), by Theorem 3 and downward closeness of $\Gamma(\mathcal{A})$, we deduce that

$$\Gamma(\mathcal{A}) \supseteq \Lambda_v^i \quad \text{iff} \quad \Gamma(\mathcal{A}(v)) \supseteq \Lambda_v^i \quad \text{iff} \quad \widetilde{v} \in \Gamma(\mathcal{A}(v)) \quad \text{iff} \quad \widetilde{v} \in \Gamma(\mathcal{A})$$

Note for all $p \in P$, $\widetilde{v}(p) \leq N_{\mathcal{A}(v)} + (B + b + 2)^{|P|+\kappa+n_e}$. Thus, by Theorem 1, we obtain the following result.

Theorem 9 *Constrained emptiness (resp., constrained universality) of lower bound automata $\mathcal{A}$ is* PSPACE*-complete, and for a given linear constraint C, it can be decided in time exponential in $|X|^2$ (resp., $|X|^4$) and in the size of the encoding of $c_{\mathcal{A}}$, and polynomial in N, $K = (B + b + 2)^{(|P|+\kappa+n_e)\cdot|P|}$, and the number of locations of $\mathcal{A}$.*

Decidability of constrained decision problems for upper bound automata For upper bound automata, the situation is dual to lower bound automata. We obtain similar results by using Theorem 6 and the fact that $\Gamma(\mathcal{A})$ is upward-closed.

Theorem 10 *Constrained universality* (*resp*., *constrained emptiness*) *of upper bound automata* $\mathcal{A}$ *is* PSPACE-*complete*, *and for a given linear constraint* C, *it can be decided in time exponential in* $|X|^2$ (*resp*., $|X|^4$) *and in the size of the encoding of* $c_{\mathcal{A}}$, *and polynomial in* N, $K = (B + b + 2)^{(|P|+\kappa+n_e)\cdot|P|}$, *and the number of locations of* $\mathcal{A}$.

General case: undecidability For L/U automata we obtain the following result.

Theorem 11 *Constrained emptiness and constrained universality are undecidable for L/U automata*. *However, if we restrict ourselves to linear constraints where each equation/inequality is either over the set of lower bound parameters or over the set of upper bound parameters*, *then the problems are* PSPACE-*complete*.

Proof The emptiness problem (restricted to finite runs) for the general class of parametric timed automata is known to be undecidable [4]. Moreover, the proof in [4] can be easily adapted to show undecidability for universality and emptiness in the case of infinite runs. These negative results already hold if every parametric atomic clock constraint f has one of the restricted forms $x \prec p$ or $-x \prec -p$, where $x \in X$, $p \in P$, and $\prec \in \{\leq, <\}$ (equivalently, f has the restricted form $x \sim p$ with $\sim \in \{\leq, <, \geq, >\}$). We reduce emptiness (resp., universality) of parametric timed automata to constrained emptiness (resp., constrained universality) of L/U automata.

Given a parametric timed automaton $\mathcal{A}$ whose clock constraints satisfy the above restriction, we construct an L/U automaton $\mathcal{A}_{L/U}$ and a linear constraint C such that $\Gamma(A) = \emptyset$ (resp., $\Gamma(\mathcal{A})$ contains all parameter valuations) if and only if $\Gamma(\mathcal{A}_{L/U}) \cap \mathrm{Sol}(C) = \emptyset$ (resp., $\Gamma(\mathcal{A}_{L/U}) \supseteq \mathrm{Sol}(C)$). The construction is very simple. $\mathcal{A}_{L/U}$ and C are obtained from $\mathcal{A}$ as follows: for every parameter p of $\mathcal{A}$, we consider an atomic clock constraint $f = x \sim p$ of $\mathcal{A}$ and for every other atomic clock constraint f' of the form $f' = y \sim' p$ we replace f' with the clock constraint $y \sim' p_1$, where p_1 is a fresh parameter. Moreover, we add the atomic linear constraint $p_1 = p$ to C. It is easy to show that the construction is correct. Thus, the first part of Theorem 11 holds.

Now consider an L/U automaton $\mathcal{A}$ and a linear constraint C given as a boolean combination of atomic linear constraints where each atomic constraint is either over the set of lower bound parameters, or over the set of upper bound parameters of $\mathcal{A}$. Without loss of generality we can assume that C is a disjunction of linear constraints of the form $C_L \wedge C_U$, where C_L and C_U are conjunctive linear constraints over the set of lower bound parameters and upper bound parameters, respectively. Observe that, on the considered instance, the non-emptiness problem has positive answer if and only if $\mathrm{Sol}(C_L \wedge C_U) \cap \Gamma(\mathcal{A}) \neq \emptyset$ for some $C_L \wedge C_U$. Analogously, the universality problem has positive answer if and only if $\mathrm{Sol}(C_L \wedge C_U) \subseteq \Gamma(\mathcal{A})$ for each disjunct $C_L \wedge C_U$. Therefore, we need to show how to solve the decision problems for a single disjunct $C_L \wedge C_U$. Let $S_L = \{v_1^L, \ldots, v_q^L\}$ (resp., $S_U = \{v_1^U, \ldots, v_r^U\}$) be the set of solutions of C_L (resp., C_U) whose components are bounded by the Pottier bound $(B + b + 2)^{|P|+\kappa+n_e}$ (as defined at the beginning of this Subsection) associated with C_L (resp., C_U). Let $\mathcal{A}$ be an L/U automaton. Then, by the characterization of the solutions of a linear constraint and monotonicity properties of L/U automata it follows that:

- To check constrained non-emptiness of $\mathcal{A}$ against $C_L \wedge C_U$, it suffices to check constrained non-emptiness for some of the upper bound automata $\mathcal{A}_1^U, \ldots, \mathcal{A}_q^U$ against C_U, where for all $1 \le i \le q$, $\mathcal{A}_i^U$ is obtained from $\mathcal{A}$ by replacing each lower bound parameter p with $v_i^L(p)$. By Theorem 10, this check is PSPACE-complete.
- To check constrained universality of $\mathcal{A}$ against $C_L \wedge C_U$, it suffices to check constrained universality of the lower bound automata $\mathcal{A}_1^L, \ldots, \mathcal{A}_r^L$ against C_L, where for all $1 \le i \le r$, $\mathcal{A}_i^L$ is obtained from $\mathcal{A}$ by replacing each upper bound parameter p with $v_i^U(p)$. By Theorem 9, this check is PSPACE-complete. □

4.5 Synthesis of parameters

In this Subsection, we show that when all the parameters in the model are of the same type (i.e., either lower bound or upper bound), it is possible to compute an explicit representation of the set $\Gamma(\mathcal{A})$ by a *linear constraint* over parameters whose size is *doubly exponential* in the number of parameters. The construction is similar to that proposed in [6] for PLTL (LTL + parameters) and is based on the results of Theorems 3 and 6. Note that these results cannot be extended to unrestricted L/U automata due to undecidability of constrained emptiness and constrained universality.

Lower Bound automata Fix a lower bound automaton $\mathcal{A}$. Let $N_\mathcal{A}$ be the constant defined in Sect. 4.1, and let $v_\mathcal{A}$ be the parameter valuation assigning $N_\mathcal{A}$ to each parameter. Since $\Gamma(\mathcal{A})$ is downward closed, by Theorem 3, we know that $\Gamma(\mathcal{A})$ contains all the parameter valuations iff $v_\mathcal{A} \in \Gamma(\mathcal{A})$ (i.e., the timed automaton $\mathcal{A}_{v_\mathcal{A}}$ is non-empty). Thus, if $v_\mathcal{A} \in \Gamma(\mathcal{A})$, then the set $\Gamma(\mathcal{A})$ is represented by the formula TRUE. Now, assume that $v_\mathcal{A} \notin \Gamma(\mathcal{A})$. Since $\Gamma(\mathcal{A})$ is downward-closed, for each parameter valuation v such that $v \ge v_\mathcal{A}$, it holds that $v \notin \Gamma(\mathcal{A})$. Thus, we need to consider only those parameter valuations that assign a value less than $N_\mathcal{A}$ to at least one parameter. We consider each parameter p and each natural number $c < N_\mathcal{A}$ in turn. Setting p to c gives us a lower bound automaton $\mathcal{A}'$ with one parameter less, and we compute the representation of $\Gamma(\mathcal{A}')$ by a recursive call. Thus, the algorithm is represented by the following recursive procedure solve($\mathcal{A}$), which returns a representation of $\Gamma(\mathcal{A})$ given by a boolean combination of atomic linear constraints of the form $p = c$. In this procedure, $P_\mathcal{A}$ denotes the set of parameters occurring in $\mathcal{A}$, and for a parameter p and constant $c \in \mathbb{N}$, $\mathcal{A}[p \leftarrow c]$ denotes the lower bound automaton obtained by replacing each occurrence of parameter p in $\mathcal{A}$ with the constant c. Also, if $P_\mathcal{A} = \emptyset$ (i.e., $\mathcal{A}$ is a TA), we denote with $\mathcal{A} \neq \emptyset$ the constant TRUE if the TA $\mathcal{A}$ has an accepting infinite run, and the constant FALSE, otherwise. The procedure solve($\mathcal{A}$) is as follows:

solve($\mathcal{A}$):
if ($P_\mathcal{A} = \emptyset$) **then return** $\mathcal{A} \neq \emptyset$
if ($v_\mathcal{A} \in \Gamma(\mathcal{A})$) **then return** TRUE

$$\textbf{return} \bigvee_{p \in P_\mathcal{A}} \bigvee_{c=0}^{N_\mathcal{A}-1} p = c \wedge \text{solve}(\mathcal{A}[p \leftarrow c])$$

Let $F(\rho_\mathcal{A}, N_\mathcal{A})$ denote the size of the representation returned by solve($\mathcal{A}$), where $\rho_\mathcal{A}$ denotes the number of parameters occurring in $\mathcal{A}$, i.e., the size of $P_\mathcal{A}$. For $\rho_\mathcal{A} = 0$, we have that $F(\rho_\mathcal{A}, N_\mathcal{A}) = 1$. Now, let $\rho_\mathcal{A} > 0$. Note that there are $\rho_\mathcal{A} N_\mathcal{A}$ nested calls to solve. Let $\mathcal{A}'$ be the lower bound automaton associated with any of these nested calls. Note that $c_{\mathcal{A}'} \le c_\mathcal{A} N_\mathcal{A}$. Moreover, by definition of $N_\mathcal{A}$ it holds that $N_\mathcal{A} = O(|Q| \cdot c_\mathcal{A}^{O(|X|^2)})$. Hence,

$N_{\mathcal{A}'} = N_{\mathcal{A}}^{O(|X|^2)}$. Thus, for $\rho_{\mathcal{A}} > 0$, we obtain the following:

$$F(\rho_{\mathcal{A}}, N_{\mathcal{A}}) \;=\; O\big(\rho_{\mathcal{A}} N_{\mathcal{A}} \cdot [\log(N_{\mathcal{A}}) + F(\rho_{\mathcal{A}} - 1, N_{\mathcal{A}}^{O(|X|^2)})]\big)$$

One can verify that this leads to a doubly exponential-bound: $F(\rho_{\mathcal{A}}, N_{\mathcal{A}}) = N_{\mathcal{A}}^{|X|^{O(\rho_{\mathcal{A}})}}$.

Theorem 12 *Given a lower bound automaton $\mathcal{A}$, one can compute a linear constraint $C_{\mathcal{A}}$ such that* $\mathrm{Sol}(C_{\mathcal{A}}) = \Gamma(\mathcal{A})$ *and the atomic constrains in $C_{\mathcal{A}}$ are of the form $p = c$ for some parameter p and constant c. Moreover, the size of $C_{\mathcal{A}}$ is doubly exponential in the number of parameters occurring in $\mathcal{A}$.*

It remains open whether one can construct a linear constraint representing $\Gamma(\mathcal{A})$ with a single exponential blow-up.

Upper bound automata The construction of a representation of the set $\Gamma(\mathcal{A})$ for a given upper bound automaton $\mathcal{A}$, is similar to that used for lower bound automata. Fix an upper bound automaton $\mathcal{A}$. Let $N_{\mathcal{A}}$ be the constant defined in Sect. 4.2, and let $v_{\mathcal{A}}$ be the parameter valuation assigning $N_{\mathcal{A}}$ to each parameter. Since $\Gamma(\mathcal{A})$ is upward closed, by Theorem 6, $\Gamma(\mathcal{A})$ is not empty iff $v_{\mathcal{A}} \in \Gamma(\mathcal{A})$. Thus, if $v_{\mathcal{A}} \notin \Gamma(\mathcal{A})$, then the set $\Gamma(\mathcal{A})$ is represented by the formula FALSE. Now, assume that $v_{\mathcal{A}} \in \Gamma(\mathcal{A})$. Since $\Gamma(\mathcal{A})$ is upward-closed, for each parameter valuation v such that $v \geq v_{\mathcal{A}}$, it holds that $v \in \Gamma(\mathcal{A})$. It remains to consider only those parameter valuations that assign a value less than $N_{\mathcal{A}}$ to at least one parameter. Here, we proceed as for lower bound automata. Thus, the algorithm is represented by the following recursive procedure solve($\mathcal{A}$), which returns a representation of $\Gamma(\mathcal{A})$ given by a boolean combination of atomic linear constraints of the form $p \geq c$. In this procedure, $P_{\mathcal{A}}$ denotes the set of parameters occurring in $\mathcal{A}$, and for a parameter p and constant $c \in \mathbb{N}$, $\mathcal{A}[p \leftarrow c]$ denotes the lower bound automaton obtained by replacing each occurrence of parameter p in $\mathcal{A}$ with the constant c. Also, if $P_{\mathcal{A}} = \emptyset$ (i.e. $\mathcal{A}$ is a TA) we denote with $\mathcal{A} \neq \emptyset$ the constant TRUE if $\mathcal{A}$ has an accepting infinite run, and the constant FALSE, otherwise. The procedure solve($\mathcal{A}$) is as follows:

solve($\mathcal{A}$):
if ($P_{\mathcal{A}} = \emptyset$) **then return** $\mathcal{A} \neq \emptyset$
if ($v_{\mathcal{A}} \notin \Gamma(\mathcal{A})$) **then return** FALSE
return $\left(\bigwedge_{p \in P_{\mathcal{A}}} p \geq N_{\mathcal{A}}\right) \vee \bigvee_{p \in P_{\mathcal{A}}} \bigvee_{c=0}^{N_{\mathcal{A}}-1} p \geq c \wedge \mathrm{solve}(\mathcal{A}[p \leftarrow c])$

As for lower bound automata, we can easily verify that the size of the representation returned by the algorithm is doubly exponential in the number of parameters occurring in $\mathcal{A}$. Thus, we obtain the following result.

Theorem 13 *Given an upper bound automaton $\mathcal{A}$, one can compute a linear constraint $C_{\mathcal{A}}$ such that* $\mathrm{Sol}(C_{\mathcal{A}}) = \Gamma(\mathcal{A})$ *and the atomic constrains in $C_{\mathcal{A}}$ are of the form $p \geq c$ for some parameter p and constant c. Moreover, the size of $C_{\mathcal{A}}$ is doubly exponential in the number of parameters occurring in $\mathcal{A}$.*

5 Parametric dense-time linear temporal logic

In this section, we define a parametric extension of the logic $\mathsf{MITL}_{0,\infty}$ [5] where the bounds of the intervals constraining the temporal operators are represented by linear expressions. However, we impose a restriction on the use of parameters similar to that imposed on the parameters of L/U automata (note that by [6], if we remove this restriction, then basic decision problems are undecidable). For such a logic, we study the related satisfiability and model-checking (w.r.t. L/U automata) problems.

5.1 Syntax and semantics of Parametric $\mathsf{MITL}_{0,\infty}$ ($\mathsf{PMITL}_{0,\infty}$)

We fix two disjoint finite sets of parameters U and L. We denote with μ (resp., λ) a linear expression over parameters $U \cup L$ such that each parameter from U (resp., L) occurs positively and each parameter from L (resp., U) occurs negatively. The set of $\mathsf{PMITL}_{0,\infty}$ formulas φ over a finite set AP of atomic propositions with upper bound parameters in U and lower bound parameters in L is defined as follows:

$$\varphi := a|\neg a|\varphi \wedge \varphi|\varphi \vee \varphi|\varphi\, \mathcal{U}_{\prec\mu}\, \varphi|\varphi\, \mathcal{U}_{\succ\lambda}\, \varphi|\varphi\, \mathcal{R}_{\prec\lambda}\, \varphi|\varphi\, \mathcal{R}_{\succ\mu}\, \varphi.$$

where $a \in AP$, $\prec \in \{\leq, <\}$, $\succ \in \{\geq, >\}$, $\mathcal{U}_{\prec\mu}$ and $\mathcal{U}_{\succ\lambda}$ are the parameterized versions of the *until* modality, and $\mathcal{R}_{\prec\lambda}\, \varphi$ and $\mathcal{R}_{\succ\mu}$ are the parameterized versions of the *release* modality. $\mathsf{MITL}_{0,\infty}$ formulas correspond to $\mathsf{PMITL}_{0,\infty}$ formulas which does not contain occurrences of parameters. In the following, for a $\mathsf{PMITL}_{0,\infty}$ formula φ, we denote with K_φ the largest absolute value of the coefficients in the linear expressions of φ plus one, and with N_φ the number of distinct subformulas of φ.

$\mathsf{PMITL}_{0,\infty}$ formulas are interpreted over timed sequences. A *timed sequence* over AP is an infinite sequence $\alpha = (\sigma_0, I_0)(\sigma_1, I_1)\cdots$ satisfying the following:

- for all i, $\sigma_i \in 2^{AP}$ and I_i is a non-empty interval in $\mathbb{R}_{\geq 0}$;
- for all i, $I_i \cap I_{i+1} = \emptyset$ and the lower bound of I_{i+1} equals the upper bound of I_i;
- each real number $t \geq 0$ belongs to some interval I_i.

For each $t \geq 0$, we denote with $\alpha(t)$ the unique σ_i such that $t \in I_i$. Two timed sequences α_1 and α_2 over AP are equivalent if for each $t \geq 0$, $\alpha_1(t) = \alpha_2(t)$.

For a formula φ, a timed sequence $\alpha = (\sigma_0, I_0)(\sigma_1, I_1)\cdots$, a parameter valuation v, and $t \in \mathbb{R}_{\geq 0}$, the satisfaction relation $(\alpha, v, t) \models \varphi$ *under valuation* v is defined as follows (we omit the clauses for boolean connectives, which are standard).

- $(\alpha, v, t) \models a \;\Leftrightarrow\; a \in \alpha(t)$;
- $(\alpha, v, t) \models \varphi\, \mathcal{U}_{\prec\mu}\, \psi \;\Leftrightarrow\;$ for some $t' > t$ such that $t' \prec \mu[v] + t$, $(\alpha, v, t') \models \psi$ and $(\alpha, v, t'') \models \varphi$ for all $t < t'' < t'$;
- $(\alpha, v, t) \models \varphi\, \mathcal{U}_{\succ\lambda}\, \psi \;\Leftrightarrow\;$ for some $t' > t$ such that $t' \succ \lambda[v] + t$, $(\alpha, v, t') \models \psi$ and $(\alpha, v, t'') \models \varphi$ for all $t < t'' < t'$;
- $(\alpha, v, t) \models \varphi\, \mathcal{R}_{\prec\lambda}\, \psi \;\Leftrightarrow\;$ for all $t' > t$ such that $t' \prec \lambda[v] + t$, either $(\alpha, v, t') \models \psi$, or $(\alpha, v, t'') \models \varphi$ for some $t < t'' < t'$;
- $(\alpha, v, t) \models \varphi\, \mathcal{R}_{\succ\mu}\, \psi \;\Leftrightarrow\;$ for all $t' > t$ such that $t' \succ \mu[v] + t$, either $(\alpha, v, t') \models \psi$, or $(\alpha, v, t'') \models \varphi$ for some $t < t'' < t'$.

For a formula φ, a timed sequence α, and a parameter valuation v, α *satisfies* φ *under valuation* v if $(\alpha, v, 0) \models \varphi$. Note that we have defined $\mathsf{PMITL}_{0,\infty}$ formulas in positive normal form. Note that $\mathsf{PMITL}_{0,\infty}$ formulas cannot distinguish equivalent timed sequences α_1 and α_2. Thus, for each $\mathsf{PMITL}_{0,\infty}$ formula φ and parameter valuation v, $(\alpha_1, v, 0) \models \varphi$ iff $(\alpha_2, v, 0) \models \varphi$.

Remark 1 Note that the until and the release operators are dual. In particular, given a $\mathsf{PMITL}_{0,\infty}$ formula φ with upper bound (resp., lower bound) parameters in U (resp., L), one can construct in linear time a $\mathsf{PMITL}_{0,\infty}$ formula ψ with upper bound (resp., lower bound) parameters in L (resp., U) such that $\neg\varphi$ and ψ are semantically equivalent.

We use $\Diamond_{\prec\mu}\,\varphi$ as an abbreviation for *true* $\mathcal{U}_{\prec\mu}\,\varphi$, and $\Box_{\prec\lambda}\,\varphi$ as an abbreviation for *false* $\mathcal{R}_{\prec\lambda}\,\varphi$.

In the following example we give a typical property that can be stated in $\mathsf{PMITL}_{0,\infty}$.

Example 2 Consider the $\mathsf{PMITL}_{0,\infty}$ formula $\varphi = \Box_{\geq 0}\,(a \rightarrow \Diamond_{\leq l}\, b)$, which represents a parametric version of the usual time response property: "whenever a holds true then b should holds true within time l". Compared to the usual time response property, here we do not need to specify a constant for the required delay, and we can instead use a parameter to express it. We can then perform the analysis for any possible value of the constant and leave to a later time the task of determining the exact constant. For example, consider the timed sequence

$$\alpha = (\{a\}, [0, 2[)\,(\{b\}, [2, 4[)\,(\emptyset, [4, 5[)\,(\emptyset, [5, 6[)\ldots$$

where all atomic propositions hold false starting from time 4. We can determine that the set of parameter valuations v such that α satisfies φ under v is specified by the constraint $l \geq 2$, and thus reply to several questions on the possible values of the parameter, such as what is the minimum constant for the property to hold.

Observe that linear expressions of the type λ and μ may evaluate to a negative value. In such cases, according to the $\mathsf{PMITL}_{0,\infty}$ semantics, formulas of the form $\varphi\,\mathcal{U}_{\prec\mu}\,\psi$ are equivalent to *false*, formulas of the form $\varphi\,\mathcal{R}_{\prec\lambda}\,\psi$ are equivalent to *true*, and constraints of the form $\succ\lambda$ turn out to be equivalent to > 0. The following lemma shows that without loss of generality, we can restrict to the parameterized operators $\Diamond_{\prec\mu}$ and $\Box_{\prec\lambda}$, along with ordinary $\mathsf{MITL}_{0,\infty}$ operators.

Lemma 5 *Given a* $\mathsf{PMITL}_{0,\infty}$ *formula* φ, *one can construct an equivalent* $\mathsf{PMITL}_{0,\infty}$ *formula* ψ *such that* $K_\psi = K_\varphi$, $N_\psi = O(N_\varphi)$, *and each temporal operator occurring in* ψ *has one of the following forms*: *or* $\Diamond_{\prec\mu}$ *or* $\Box_{\prec\lambda}$ *or* $\mathcal{U}_{>0}$ *or* $\mathcal{R}_{>0}$.

Proof The result directly follows from the following equivalences, where $\xi = \varphi\,\mathcal{U}_{>0}\,\psi$ and $\xi' = \varphi\,\mathcal{R}_{>0}\,\psi$:

$$\varphi\,\mathcal{U}_{\prec\mu}\,\psi \equiv \varphi\,\mathcal{U}_{>0}\,\psi \wedge \Diamond_{\prec\mu}\,\psi$$

$$\varphi\,\mathcal{U}_{>\lambda}\,\psi \equiv \Box_{\leq\lambda}\,(\varphi \;\wedge\; \varphi\,\mathcal{U}_{>0}\,\psi) \;\wedge\; \xi$$

$$\varphi\,\mathcal{U}_{\geq\lambda}\,\psi \equiv \Box_{<\lambda}\,\varphi \;\wedge\; \Box_{\leq\lambda}\,(\psi \vee (\varphi \;\wedge\; \varphi\,\mathcal{U}_{>0}\,\psi)) \;\wedge\; \xi$$

$$\varphi\,\mathcal{R}_{\prec\lambda}\,\psi \equiv (\varphi\,\mathcal{R}_{>0}\,\psi) \vee \Box_{\prec\lambda}\,\psi$$

$$\varphi\,\mathcal{R}_{>\mu}\,\psi \equiv \Diamond_{\leq\mu}\,(\varphi \;\vee\; \varphi\,\mathcal{R}_{>0}\,\psi) \;\vee\; \xi'$$

$$\varphi\,\mathcal{R}_{\geq\mu}\,\psi \equiv \Diamond_{<\mu}\,\varphi \;\vee\; \Diamond_{\leq\mu}\,(\psi \;\wedge\; (\varphi \;\vee\; \varphi\,\mathcal{R}_{>0}\,\psi)) \;\vee\; \xi'$$

The formulas ξ and ξ' are added to ensure the equivalence when the linear expression in the subscripts evaluates to a negative value. Here, we prove the second equivalence (the other equivalences can be proved similarly). Fix a parameter valuation v, a timed sequence α, and $t \geq 0$.

First, assume that $(\alpha, v, t) \models \varphi\, \mathcal{U}_{>\lambda}\, \psi$. From the semantics of the until modality, we get that there is $t' > t$ such that $t' > t + \lambda[v]$, $(\alpha, v, t') \models \psi$ and $(\alpha, v, t'') \models \varphi$ for all $t < t'' < t'$. This clearly implies that $(\alpha, v, t) \models \xi$. Moreover, for all t'' such that $t'' > t$ and $t'' \leq t + \lambda[v]$, we have that $(\alpha, v, t'') \models \varphi$ and $(\alpha, v, t'') \models \varphi\, \mathcal{U}_{>0}\, \psi$ (recall $t' > t + \lambda[v]$). Hence, $(\alpha, v, t) \models \Box_{\leq\lambda}\, (\varphi \,\wedge\, \varphi\, \mathcal{U}_{>0}\, \psi) \,\wedge\, \xi$.

Now, assume that $(\alpha, v, t) \models \Box_{\leq\lambda}\, (\varphi \,\wedge\, \varphi\, \mathcal{U}_{>0}\, \psi) \,\wedge\, \xi$. Since $\xi = \varphi\, \mathcal{U}_{>0}\, \psi$, if $\lambda[v] \leq 0$, then $(\alpha, v, t) \models \varphi\, \mathcal{U}_{>\lambda}\, \psi$. Now, assume that $\lambda[v] > 0$ and let $t_0 = t + \lambda[v]$. Since $(\alpha, v, t) \models \Box_{\leq\lambda}\, (\varphi \,\wedge\, \varphi\, \mathcal{U}_{>0}\, \psi)$, it holds that for all $t < t' \leq t_0$, $(\alpha, v, t') \models \varphi$ and $(\alpha, v, t') \models \varphi\, \mathcal{U}_{>0}\, \psi$. In particular, $(\alpha, v, t_0) \models \varphi\, \mathcal{U}_{>0}\, \psi$. Therefore, there exists $t_1 > t_0$ (hence, $t_1 > t + \lambda[v]$) such that $(\alpha, v, t_1) \models \psi$ and $(\alpha, v, t'') \models \varphi$ for all $t < t'' < t_1$. Hence, $(\alpha, v, t) \models \varphi\, \mathcal{U}_{>\lambda}\, \psi$. Thus, we can conclude that the second equivalence holds. □

5.2 Decision problems

In this subsection, for the logic $\mathsf{MITL}_{0,\infty}$, we study the related satisfiability and model-checking (w.r.t. L/U automata) problems. First, we need additional definitions.

In the following, we assume that each L/U automaton $\mathcal{A}$ is equipped with a labeling function $L_{\mathcal{A}} : Q \to 2^{AP}$ assigning to each location a set of propositions in AP. For each parameter valuation v and infinite run $\pi = s_0 \xrightarrow{\delta_0,\tau_0} s_1 \xrightarrow{\delta_1,\tau_1} s_2 \cdots$ of $\mathcal{A}_v$, we associate to π a set of *equivalent* timed sequences over AP as follows. Let $s_i = (q_i, w_i)$ for each i, and let $(s_0, t_0) \xrightarrow{\delta_0,\tau_0} (s_1, t_1) \xrightarrow{\delta_1,\tau_1} (s_2, t_2) \cdots$ be the *extended* infinite run associated with π such that $t_0 = 0$. The run π induces the mapping $\pi^* : \mathbb{R}_{\geq 0} \to Q$ defined as:

- for each $t \geq 0$, $\pi^*(t) = q_j$, where j is the smallest index i such that $t_i \leq t < t_{i+1}$.

A timed sequence α over AP is *generated* by the run π if for each $t \geq 0$, $\alpha(t) = L_{\mathcal{A}}(\pi^*(t))$. A timed sequence α is *accepted* by the TA $\mathcal{A}_v$ if α is generated by some infinite accepting run of $\mathcal{A}_v$ starting from the initial state. Note that if a timed sequence α is accepted by $\mathcal{A}_v$, then each timed sequence which is equivalent to α is accepted by $\mathcal{A}_v$ too. In the following we denote by $\Pi_+(U, L)$ (resp., $\Pi_-(U, L)$) the class of L/U automata such that the lower bound parameters are from L (resp., U) and the upper bound parameters are from U (resp., L). For a given $\mathsf{PMITL}_{0,\infty}$ formula φ with lower (resp., upper) bound parameters from L (resp., U), an L/U automaton $\mathcal{A}_+ \in \Pi_+(U, L)$, and an L/U automaton $\mathcal{A}_- \in \Pi_-(U, L)$, we consider the emptiness and universality problems for the following sets of parameter valuations:

- the set $S(\varphi)$ of parameter valuations v that make φ satisfiable (i.e. such that there is a timed sequence that satisfies φ under valuation v);
- the set $V(\varphi)$ of parameter valuations v that make φ valid (i.e. such that each timed sequence satisfies φ under valuation v);
- the set $S(\mathcal{A}_+, \varphi)$ of parameter valuations v for which there exists a timed sequence accepted by $(\mathcal{A}_+)_v$ that satisfies φ under valuation v;
- the set $V(\mathcal{A}_-, \varphi)$ of parameter valuations v for which every timed sequence accepted by $(\mathcal{A}_-)_v$ satisfies φ under valuation v.

Note that $V(\mathcal{A}, \varphi)$ represents the set of parameter valuations v for which $\mathcal{A}_v$ *satisfies* φ according to the usual definition of the linear-time model-checking problem. Moreover, note that $V(\varphi)$ is the complement of $S(\neg\varphi)$ and $V(\mathcal{A}, \varphi)$ is the complement of $S(\mathcal{A}, \neg\varphi)$.

We solve the considered decision problems by reducing them to corresponding problems on L/U automata. The key of these reductions is the translation of $\mathsf{PMITL}_{0,\infty}$ formulas into equivalent L/U automata. Such a translation relies on the result given in [5] concerning $\mathsf{MITL}_{0,\infty}$ and timed automata, which we recall.

Theorem 14 [5] *Let φ be a $\mathsf{MITL}_{0,\infty}$ formula. Then, one can construct a timed automaton $\mathcal{A}_\varphi$ such that $\mathcal{A}_\varphi$ accepts a timed sequence α if and only if α is a model of φ. Moreover, $\mathcal{A}_\varphi$ has $O(2^{N_\varphi})$ locations, $O(N_\varphi)$ clocks and constants bounded by K_φ.*

Now, we can prove the following result.

Theorem 15 *Given a $\mathsf{PMITL}_{0,\infty}$ formula φ with lower (resp., upper) bound parameters in L (resp., U), one can construct an L/U automaton $\mathcal{A}_\varphi \in \Pi_+(U, L)$ such that for each parameter valuation v: $(\mathcal{A}_\varphi)_v$ accepts a timed sequence α if and only if $(\alpha, v, 0) \models \varphi$. Moreover, $\mathcal{A}_\varphi$ has $O(2^{N_\varphi})$ locations, $O(N_\varphi)$ clocks, and constants bounded by K_φ.*

Proof Fix a $\mathsf{PMITL}_{0,\infty}$ formula φ with lower (resp., upper) bound parameters in L (resp., U). By [5], for each parameter valuation v and timed sequence α, there is a timed sequence $\alpha_{\varphi,v}$ equivalent to α such that for each interval I of $\alpha_{\varphi,v}$ and subformula ψ of φ, the following holds: for all $t, t' \in I$, $(\alpha, v, t) \models \psi$ iff $(\alpha, v, t') \models \psi$. Thus, we can restrict ourselves to consider only this class of timed sequences $\alpha_{\varphi,v}$ (for a given parameter valuation v). Moreover, according to Lemma 5, we can assume that for the given $\mathsf{PMITL}_{0,\infty}$ formula φ, the only parametric subformulas are of the form $\Diamond_{\prec\mu}\psi$ and $\Box_{\prec\lambda}\psi$. This suggests to use the construction given in [5] in order to obtain a PTA that accepts the models of a $\mathsf{PMITL}_{0,\infty}$ formula for each parameter valuation. We only need to describe the additions that are needed to handle the parametric subformulas.

Fix a parameter valuation v. Consider first a subformula ψ of φ of the form $\Diamond_{\prec\mu}\theta$. For checking the fulfillment of ψ, we need to use one clock, say x, along with the clock constraint $x \prec \mu$. As in [5], each location of $\mathcal{A}$ keeps track of the set of subformulas of φ which hold at the current time (w.r.t. the parameter valuation v). In order to witness the fulfillment of ψ at the current time t, the automaton resets x and stores (in its finite control) the obligation that θ must hold at a time $t + d$ such that $0 < d \prec \mu[v]$. The obligation is discharged as soon as θ is fulfilled (i.e., an appropriate θ-state is found). Clearly, if an obligation for ψ is already pending (and cannot be discharged), we cannot reset x otherwise we would not be able to check the pending obligation. However, this is not needed since a witness for the previous obligation will also prove the fulfillment of ψ at the current time. Once the obligation is discharged, the clock x can be reused. Thus, one clock suffices to check the subformula ψ as often as necessary. Moreover, each transition whose source is a location q that stores an obligation for ψ (that cannot be discharged locally), uses an atomic clock constraint $x \prec \mu$ as a conjunct of the associated clock constraint. This ensures that an obligation for ψ is not mistakenly discharged because of a witness at a time $t + d$ with $d \not\prec \mu[v]$. Observe that the above behavior needs to be implemented over intervals and there are some subtleties concerning the treatment of open intervals that have been omitted here. The details on these aspects do not differ from the case of formulas of the form $\Diamond_{\prec c}\varphi$, where c is a constant, which have been carefully explained in [5].

Subformulas of φ of the form $\Box_{\prec\lambda}\theta$ are dual to formulas of the form $\Diamond_{\prec\mu}\theta$. Therefore, we can argue similarly that an automaton can check for the fulfillment of one of such formulas, simply using a clock x and a clock constraint $-x < -\lambda$, if $\prec$ is $\leq$, and $-x \leq -\lambda$, otherwise (the linear expression $-\lambda$ is defined in the obvious way).

For a $\mathsf{PMITL}_{0,\infty}$ formula φ, let $\mathcal{A}_\varphi$ be the parametric timed automaton obtained with the above sketched construction. Note that the only parameters used in $\mathcal{A}_\varphi$ appear in φ and thus belong to $L \cup U$. Moreover, each parameter of φ from L is a lower bound parameter for $\mathcal{A}_\varphi$ and each parameter of φ from U is an upper bound parameter for $\mathcal{A}_\varphi$. Therefore, $\mathcal{A}_\varphi$ is an L/U automaton in $\Pi_+(U, L)$. □

For an L/U automaton $\mathcal{A}$, let $n_\mathcal{A}$ be the number of its locations, $c_\mathcal{A}$ the largest absolute value of the coefficients in its linear expressions, and $X_\mathcal{A}$ be its set of clocks.

Lemma 6 *Given two L/U automata $\mathcal{A}_1, \mathcal{A}_2 \in \Pi_+(U, L)$ over the set of propositions AP, one can construct an L/U automaton $\mathcal{A} \in \Pi_+(U, L)$ over AP such that for each parameter valuation v, $\mathcal{A}_v$ accepts the intersection of the languages of timed sequences accepted by $(A_1)_v$ and $(A_2)_v$. Moreover, $n_\mathcal{A} = O(n_{A_1} \cdot n_{A_2})$, $|X_\mathcal{A}| = |X_{\mathcal{A}_1}| + |X_{\mathcal{A}_2}|$, and $c_\mathcal{A} = max\{c_{\mathcal{A}_1}, c_{\mathcal{A}_2}\}$.*

Proof By a trivial readaptation of the standard product construction for Büchi timed automata [1]. □

Now, we can prove the main result of this section.

Theorem 16 *For a $\mathsf{PMITL}_{0,\infty}$ formula φ with lower (resp., upper) bound parameters from L (resp., U), an L/U automaton $\mathcal{A}_+ \in \Pi_+(U, L)$, and an L/U automaton $\mathcal{A}_- \in \Pi_-(U, L)$, checking for emptiness and universality of the sets $S(\varphi)$, $V(\varphi)$, $S(\mathcal{A}_+, \varphi)$, and $V(\mathcal{A}_-, \varphi)$ is* PSPACE*-complete.*

Proof Recall that by Remark 1, the formula $\neg\varphi$ can be converted in linear time into an equivalent $\mathsf{PMITL}_{0,\infty}$ formula ψ with lower bound parameters from U and upper bound parameters from L. Also, $V(\varphi)$ is the complement of $S(\psi)$ and $V(\mathcal{A}, \varphi)$ is the complement of $S(\mathcal{A}, \psi)$ for each L/U automaton $\mathcal{A}$. Moreover, we have that $\mathcal{A} \in \Pi_+(L, U)$ iff $\mathcal{A} \in \Pi_-(U, L)$. Thus, we just need to show that checking for emptiness and universality of the sets $S(\varphi)$ and $S(\mathcal{A}, \varphi)$ is PSPACE-complete for any L/U automaton $\mathcal{A} \in \Pi_+(U, L)$ and $\mathsf{PMITL}_{0,\infty}$ formula φ with lower bound parameters from L and upper bound parameters from U.

Let us consider the set $S(\mathcal{A}, \varphi)$ (the proof for the set $S(\varphi)$ is simpler), which consists of the parameter valuations v such that there is timed sequence α accepted by $\mathcal{A}_v$ and satisfying $(\alpha, v, 0) \models \varphi$. By Theorem 15 and Lemma 6, we can construct a L/U automaton $\mathcal{B} \in \Pi_+(U, L)$ such that for each parameter valuation v and timed sequence α, α is accepted by $\mathcal{B}_v$ iff α is accepted by $\mathcal{A}_v$ and $(\alpha, v, 0) \models \varphi$. Moreover, $n_\mathcal{B} = O(n_\mathcal{A} \cdot 2^{N_\varphi})$, $|X_\mathcal{B}| = O(N_\varphi + |X_\mathcal{A}|)$, and $c_\mathcal{B} = \max\{c_\mathcal{A}, K_\varphi\}$. Since $v \in \Gamma(\mathcal{B})$ iff there is some timed sequence accepted by $\mathcal{B}_v$, it follows that $\Gamma(\mathcal{B}) = S(\mathcal{A}, \varphi)$. Thus, by Theorem 8 the result follows. □

6 Conclusion

We have studied several main decision problems on L/U automata. In particular, we have shown that the emptiness, finiteness, and universality problems for the set of parameter valuations $\Gamma(\mathcal{A})$ for which there is an infinite accepting run are decidable and PSPACE-complete. This also allows us to prove decidability of a parametric extension of $\mathsf{MITL}_{0,\infty}$. Moreover, we have studied a constrained version of emptiness and universality where parameters are constrained by linear systems of equations and inequalities. Furthermore, we have shown that when all the parameters in the model are of the same type (i.e., either lower bound or upper bound), it is possible to compute an explicit representation of the set $\Gamma(\mathcal{A})$ by linear systems over parameters whose size is *doubly exponential* in the number of parameters.

For the ease of presentation we do not allow to specify clock invariants on locations of L/U automata. However, it is simple to verify that the addition of invariants would not change the validity of our arguments.

The limitations we have placed on the use of parameters in the chosen models are needed for achieving decidability, as shown by our undecidability results and similar results from the literature (see [4, 6]).

The approach we have followed to decide problems for $\mathsf{PMITL}_{0,\infty}$ does not have a direct generalization to the parametric extension of full MITL [5]. The main reason is that the translation of MITL formulas to timed automata requires a number of clocks that is related to the lengths of the intervals in the formulas. If we allow occurrences of parameters both in the left and the right bounds of an interval, applying the construction given in [5] we would need to define an automaton where the number of clocks is itself parameterized. A recent paper [10], has settled the questions about the complexity of the main decision problems of parametric MITL and compared the expressiveness of some fragments of this logic.

Recent results on decidable fragments of Metric Temporal Logic which allow singular intervals as subscripts of temporal operators [7] open a new direction on the investigation of the impact of parameters in real-time linear temporal logic. Also, expressiveness of parametric temporal logic has not been much investigated thus far and we see this as an interesting future research line. Finally, an orthogonal research direction would be to investigate the extension of our results to real-valued parameters. The results we have shown in this paper answer only partially to this problem.

Appendix

A.1 Proof of Lemma 4

Lemma 4 (Cutting Lemma) *Let v be a parameter valuation which* evaluates positive *for a* PTA $\mathcal{A}$. *Let $\pi\rho$ be a run of $\mathcal{A}_v$ from $s'_0 = (q_0, w'_0)$ with $\rho = s_0 \xrightarrow{\delta_0,\tau_0} s_1 \cdots s_{n-1} \xrightarrow{\delta_{n-1},\tau_{n-1}} s_n$ such that $s'_0 \approx_{v_{null}} s_0$,* $\mathrm{DUR}(\pi) \geq 1$, *and for each non-constant linear expression e of $\mathcal{A}$ and parametric clock $x \neq x_0$: if x is reset along π, then x is reset along ρ and $w'_0(x) +$* $\mathrm{DUR}(\pi\rho) < e[v]$.

Then, there is a finite run $\rho' = s'_0 \xrightarrow{\delta_0,\tau'_0} s'_1 \cdots s'_{n-1} \xrightarrow{\delta_{n-1},\tau'_{n-1}} s'_n$ of $\mathcal{A}_v$ such that $\mathrm{DUR}(\rho') \equiv \mathrm{DUR}(\rho)$, *and $s'_n \sqsubseteq s_n$.*

Proof Let $(s_0, t_0) \xrightarrow{\delta_0,\tau_0} (s_1, t_1) \cdots (s_{n-1}, t_{n-1}) \xrightarrow{\delta_{n-1},\tau_{n-1}} (s_n, t_n)$ be the extended run of $\mathcal{A}_v$ associated with ρ. Recall that $t_0 = 0$. Moreover, let $s_h = (q_h, w_h)$ for $0 \leq h \leq n$. We first prove the following claim.

Claim There is an extended run of $\mathcal{A}_v$ of the form $(s'_0, t'_0) \xrightarrow{\delta_0,\tau'_0} (s'_1, t'_1) \cdots (s'_{n-1}, t'_{n-1}) \xrightarrow{\delta_{n-1},\tau'_{n-1}} (s'_n, t'_n)$ such that $t'_0 = 0$, and for all $0 \leq h \leq n$, $s'_h = (q_h, w'_h)$ and $(s'_h, t'_h) \approx_{v_{null}} (s_h, t_h)$. Moreover, for all $x \in X$ and $0 \leq h < n$, $w_h(x) \equiv w'_h(x)$ implies $w_{h+1}(x) \equiv w'_{h+1}(x)$.

Proof Setting $t'_0 = 0$, by hypothesis $(s'_0, t'_0) \approx_{v_{null}} (s_0, t_0)$. Now, assume the existence of an extended finite run ρ'_k of $\mathcal{A}_v$ of the form $\rho'_k = (s'_0, t'_0) \xrightarrow{\delta_0, \tau'_0} (s'_1, t'_1) \cdots (s'_{k-1}, t'_{k-1}) \xrightarrow{\delta_{k-1}, \tau'_{k-1}} (s'_k, t'_k)$ such that $k < n$, $s'_k = (q_k, w'_k)$, and $(s'_h, t'_h) \approx_{v_{null}} (s_h, t_h)$ for $0 \leq h \leq k$. Then, it suffices to show that there is an extended edge in $[\![\mathcal{A}]\!]_v$ of the form $(s'_k, t'_k) \xrightarrow{\delta_k, \tau'_k} (s'_{k+1}, t'_{k+1})$ such that $(s'_{k+1}, t'_{k+1}) \approx_{v_{null}} (s_{k+1}, t_{k+1})$, $s'_{k+1} = (q_{k+1}, w'_{k+1})$, and for all $x \in X$, $w_{k+1}(x) \equiv w'_{k+1}(x)$ if $w_k(x) \equiv w'_k(x)$.

Since $(s_k, t_k) \xrightarrow{\delta_k, \tau_k} (s_{k+1}, t_{k+1})$ and $(s'_k, t'_k) \approx_{v_{null}} (s_k, t_k)$, the case of a time transition directly follows from Proposition 1(2). Now, assume that $\delta_k = (q_k, g, r, q_{k+1})$. By Proposition 1(1), it suffices to show that $(v, w'_k) \models g$. Since $s'_k \approx_{v_{null}} s_k$, w'_k satisfies all non-parametric atomic clock constraints occurring in g. Now, assume that f is a parametric atomic clock constraint of g of the form $x - y \prec e$. If $x = x_0$, then since $e[v] > 0$ (v evaluates positive for $\mathcal{A}$), condition $(v, w'_k) \models f$ holds. Now, assume that $x \neq x_0$. We distinguish three cases:

1. Clock x is reset along π. By hypothesis, $w'_0(x) + \mathrm{DUR}(\pi\rho) < e[v]$. Since $t_k \leq \mathrm{DUR}(\rho)$, $t'_k \equiv t_k$, $\mathrm{DUR}(\pi) \geq 1$, and $w'_k(x) \leq t'_k + w'_0(x)$, it follows that $w'_k(x) < e[v]$. Hence, $w'_k(x) - w'_k(y) \leq w'_k(x) < e[v]$, i.e. $(v, w'_k) \models f$.
2. Clock x is reset along $\rho[0, k]$ (hence, also along ρ'_k). Since $e[v] > 0$, if $w'_k(x) - w'_k(y) \leq 0$, then $(v, w'_k) \models f$. Now, assume that $w'_k(x) - w'_k(y) > 0$. Let $0 < h_1 \leq k$ such that x is reset on the edge $\rho'_k[h_1 - 1, h_1]$ and x is not reset along $\rho'_k[h_1, k]$. Since $w'_k(x) - w'_k(y) > 0$, it follows that clock y must be reset along $\rho'_k[h_1, k]$. Therefore, there is $h_1 < h_2 \leq k$ such that y is reset on the edge $\rho'_k[h_2 - 1, h_2]$ and y is not reset along $\rho'_k[h_2, k]$. Thus, $w'_k(x) - w'_k(y) = w'_{h_2}(x) - w'_{h_2}(y) = w'_{h_2}(x)$ and $w'_{h_1}(x) = 0$. By construction, x and y are not reset also along $\rho[h_2, k]$, x is reset on the edge $\rho[h_1 - 1, h_1]$ and y is reset on the edge $\rho[h_2 - 1, h_2]$. Thus, we obtain that $w_k(x) - w_k(y) = w_{h_2}(x)$ and $w_{h_1}(x) = w'_{h_1}(x) = 0$ (in particular, $w'_{h_1}(x) \equiv w_{h_1}(x)$). Since $h_2 > h_1$, by the induction hypothesis, we obtain that $w_{h_2}(x) \equiv w'_{h_2}(x)$. Thus, $w_k(x) - w_k(y) \equiv w'_k(x) - w'_k(y)$, hence $(v, w'_k) \models f$.
3. Clock x is not reset along $\pi \cdot \rho[0, k]$. In particular, $w_0(x) = w'_0(x) + \mathrm{DUR}(\pi)$. First, assume that $y \neq x_0$ and y is not reset along $\rho[0, k]$. Then, $w_k(x) - w_k(y) = w_0(x) - w_0(y)$ and $w'_k(x) - w'_k(y) = w'_0(x) - w'_0(y)$. Since $w'_0(x) = w_0(x) - \mathrm{DUR}(\pi)$ and $w_0(y) \leq w'_0(y) + \mathrm{DUR}(\pi)$, it follows that $w'_k(x) - w'_k(y) \leq w_k(x) - w_k(y)$, hence $(v, w'_k) \models f$.

   Now, assume that either $y = x_0$ or y is reset along $\rho[0, k]$ (hence, also along ρ'_k). In this case there is $0 \leq h \leq k$ such that $w_k(x) - w_k(y) = w_0(x) + t_h$ and $w'_k(x) - w'_k(y) = w'_0(x) + t'_h$. Since $w_0(x) = w'_0(x) + \mathrm{DUR}(\pi)$, we obtain that $w_k(x) - w_k(y) = w'_k(x) - w'_k(y) + \mathrm{DUR}(\pi) + (t_h - t'_h) \geq w'_k(x) - w'_k(y)$ (since $\mathrm{DUR}(\pi) \geq 1$ and $t_h \equiv t'_h$). Hence, $(v, w'_k) \models f$.

Thus, we conclude that $(v, w'_k) \models g$. □

Let $(s'_0, t'_0) \xrightarrow{\delta_0, \tau'_0} (s'_1, t'_1) \cdots (s'_{n-1}, t'_{n-1}) \xrightarrow{\delta_{n-1}, \tau'_{n-1}} (s'_n, t'_n)$ be an extended run of $\mathcal{A}_v$ satisfying the claim above with $s'_h = (q_h, w'_h)$ for all $0 \leq h \leq n$. Let ρ' be the associated run of $\mathcal{A}_v$. We have that $\mathrm{DUR}(\rho') = t'_n$ and $\mathrm{DUR}(\rho) = t_n$. By the claim above, $\mathrm{DUR}(\rho) \equiv \mathrm{DUR}(\rho')$. Thus, in order to conclude the proof of the lemma, we need to show that $s'_n \sqsubseteq s_n$. By the claim above, $s_n \approx_{v_{null}} s'_n$. Now, let $x, y \in X(P, \mathcal{A}) \cup \{x_0\}$ such that $w'_n(x) - w'_n(y) > 0$. Note that $x \neq x_0$. We distinguish two cases:

- Clocks x is never reset along ρ (hence, also along ρ' and π). By using an argument similar to Case 3 in the proof of Claim 1, we obtain that $w'_n(x) - w'_n(y) \leq w_n(x) - w_n(y)$.

- Clock x is reset along ρ. By using an argument similar to Case 2 in the proof of Claim 1, we deduce that $w_n(x) - w_n(y) \equiv w'_n(x) - w'_n(y)$.

Thus, either $w'_n(x) - w'_n(y) \leq w_n(x) - w_n(y)$ or $w_n(x) - w_n(y) \equiv w'_n(x) - w'_n(y)$. Since $s_n \approx_{v_{null}} s'_n$, we have that $s'_n \sqsubseteq s_n$. This concludes the proof of the lemma. $\square$

References

1. Alur R, Dill D (1994) A theory of timed automata. Theor Comput Sci 126(2):183–235
2. Alur R, Madhusudan P (2004) Decision problems for timed automata: a survey. In: Formal methods for the design of real-time systems. LNCS, vol 3185. Springer, Berlin, pp 1–24
3. Alur R, Courcoubetis C, Dill DL (1993) Model-checking in dense real-time. Inf Comput 104(1):2–34
4. Alur R, Henzinger ThA, Vardi MY (1993) Parametric real-time reasoning. In: Proc of the 25th ACM symposium on theory of computing (STOC'93), pp 592–601
5. Alur R, Feder T, Henzinger ThA (1996) The benefits of relaxing punctuality. J ACM 43(1):116–146
6. Alur R, Etessami K, La Torre S, Peled D (2001) Parametric temporal logic for "model measuring". ACM Trans Comput Log 2(3):388–407
7. Bouyer P, Markey N, Ouaknine J, Worrell J (2007) The cost of punctuality. In: Proc of the 22nd annual IEEE symposium on logic in computer science (LICS'07). IEEE Computer Society Press, Los Alamitos, pp 109–118
8. Bruyère V, Raskin JF (2003) Real-time model-checking: parameters everywhere. In: 23rd int conf on foundations of software technology and theoretical computer science (FSTTCS'03). LNCS, vol 2914. Springer, Berlin, pp 100–111
9. Bruyère V, Dall'Olio E, Raskin JF (2003) Durations, parametric model-checking in timed automata with presburger arithmetic. In: Proc 20th int symp theoretical aspects of computer science (STACS'03). LNCS, vol 2607. Springer, Berlin, pp 687–698
10. Di Giampaolo B, La Torre S, Napoli M (2009) Parametric metric interval temporal logic. Technical report Università di Salerno
11. Emerson EA, Trefler R (1999) Parametric quantitative temporal reasoning. In: Proc 14th ann symp logic in computer science (LICS'99). IEEE Computer Society Press, Los Alamitos, pp 336–343
12. Henzinger T, Kopke P, Puri A, Varaiya P (1998) What's decidable about hybrid automata. J Comput Syst Sci 57:94–124
13. Hune T, Romijn J, Stoelinga M, Vaandrager F (2002) Linear parametric model checking of timed automata. J Log Algebraic Program 52–53:183–220
14. IEEE Computer Society (1996) IEEE standard for a high performance serial bus. Std 1394–1995, August 1996
15. Lamport L (1987) A fast mutual exclusion algorithm. ACM Trans Comput Syst 5(1):1–11
16. Pnueli A (1977) The temporal logic of programs. In: Proc of the 18th IEEE symposium on foundations of computer science, pp 46–77
17. Pottier L (1991) Minimal solutions of linear diophantine systems: bounds and algorithms. In: Proc of 4th int conf on rewriting techniques and applications (RTA'91). LNCS, vol 488. Springer, Berlin, pp 162–173
18. Wang F (1996) Parametric timing analysis for real-time systems. Inf Comput 130(2):131–150