Decision Problems for Lower/Upper Bound Parametric Timed Automata*

Laura Bozzelli[1] and Salvatore La Torre[2]

[1] Università di Napoli Federico II , Via Cintia, 80126 - Napoli, Italy
[2] Università degli Studi di Salerno, Via Ponte Don Melillo - 84084 Fisciano, Italy

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 un upper bound. For such automata, we show that checking if for a parameter valuation (resp., all parameter valuations) there is an infinite accepting run 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 to the upper bound parameters in the linear system, and are undecidable in general. Finally, we consider a parametric extension of $\text{MITL}_{0,\infty}$, and prove that the related satisfiability and model checking (w.r.t. L/U automata) problems are PSPACE-complete.

1 Introduction

Timed automata [2] 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 automata have been intensively studied by many authors, and significant progresses have been done in developing verification algorithms, heuristics, and tools (see [6] 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 [5], 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

* This research was partially supported by the MIUR grant ex-60% 2005-2006 Università di Salerno, and the European Commission via FP6 program under contracts FP6-1596 AEOLUS.

L. Arge et al. (Eds.): ICALP 2007, LNCS 4596, pp. 925–936, 2007.

related to various hard and open problems of logic and automata theory [5]. In [11], 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 [13], the root contention protocol [12] and other known examples from the literature (see [11]). 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.

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 and universality (i.e., if $\Gamma(\mathcal{A})$ contains all the parameter valuations) 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 either upper bounds or lower 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 constraint.

An important consequence of our results on L/U automata is the extension to the dense-time paradigm of the results shown in [3]. We define a parametric extension of the temporal logic $\text{MITL}_{0,\infty}$ [4], denoted $\text{PMITL}_{0,\infty}$, 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.

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 [3] where the authors study the logic LTL [14] 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 [16,9] where decidability is shown for logics obtained as extensions of TCTL [1] with parameters. In [7], 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 [10]. 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 [10].

Due to the lack of space, for the omitted details we refer the interested reader to a forthcoming extended version of this paper.

2 Parametric Timed 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 $c_0 + c_1 p_1 + \ldots + c_m p_m$ 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. The *null parameter valuation*, denoted v_{null}, is the valuation assigning 0 to each parameter. For the linear expression e above, $e[v]$ denotes the integer $c_0 + c_1 v(p_1) + \ldots + c_m v(p_m)$.

We fix a finite set of *clocks* X. For the ease of presentation, 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 s.t. $w(x_0) = 0$. For a 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. We denote by $X(P)$ 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 $y - x \prec e$. A *PTA* $\mathcal{A}$ is called a *timed automaton* (*TA*, for short), if $\mathcal{A}$ does not contain occurrences

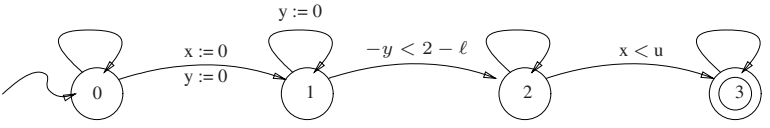

Fig. 1. An L/U automaton

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 $[\![\mathcal{A}]\!]_v$, is the labelled transition system $\langle S, \twoheadrightarrow\rangle$ over $(\Delta \cup \{\bot\}) \times \mathbb{R}_{\geq 0}$, where S is the set of $\mathcal{A}$ states and for $\tau \geq 0$, $(q, w) \xrightarrow{\delta,\tau} (q', w')$ iff:

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

An *infinite run* of $[\![\mathcal{A}]\!]_v$ is an infinite path $\rho = s_0 \xrightarrow{\delta_0,\tau_0} s_1 \xrightarrow{\delta_1,\tau_1} s_2 \ldots$ 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 $\rho = s_0 \xrightarrow{\delta_0,\tau_0} s_1 \ldots s_{n-1} \xrightarrow{\delta_{n-1},\tau_{n-1}} s_n$ of $[\![\mathcal{A}]\!]_v$. The *duration* of ρ, denoted by $DUR(\rho)$, is defined as $DUR(\rho) = \sum_{i=0}^{i=n-1} \tau_i$. We denote with $\Gamma(\mathcal{A})$ the set of parameter valuations v such that there exists an accepting infinite run of $[\![\mathcal{A}]\!]_v$ from the initial state $(q^0, \overrightarrow{0})$.

Given a linear expression $e = c_0 + c_1 p_1 + \ldots c_m p_m$ and a parameter $p_i \in P$, we say that p_i *occurs positively in* e if $c_i \geq 0$. Analogously, we say that p_i *occurs negatively in* e if $c_i \leq 0$. A *lower bound parameter* (resp., an *upper bound parameter*) of a *PTA* $\mathcal{A}$ is a parameter that only occurs negatively (resp., occurs positively) in the expressions of $\mathcal{A}$. 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. 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 is easy to verify that* $[\![\mathcal{A}]\!]_v$ *has an infinite run from location* 0 *visiting infinitely often location* 3 *iff* $v(\ell) < v(u) + 2$ *and* $v(u) > 0$*. Therefore,* $\Gamma(\mathcal{A}) = \{v \mid v(\ell) < v(u) + 2 \text{ and } v(u) > 0\}$.

For an L/U automaton $\mathcal{A}$, we consider the following decision problems on $\Gamma(\mathcal{A})$:

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

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

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

Let $\mathcal{A} = \langle Q, q^0, \Delta, F \rangle$ be a *PTA* and v be a parameter valuation. We denote by K_v 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 over $\mathcal{A}$ states defined as: $(q, w) \approx_v (q', w')$ iff $q = q'$ and for all clocks $x, y \in X$, (i) $w(x) - w(y) \geq 0$ iff $w'(x) - w'(y) \geq 0$, (ii) $|w(x) - w(y)| \approx_{K_v} |w'(x) - w'(y)|$, (iii) $fract(w(x)) \leq fract(w(y))$ iff $fract(w'(x)) \leq fract(w'(y))$ (*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 $O(|Q| \cdot (2K_v + 2)^{|X|^2})$ [2] (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 emptiness of the finite–state quotient graph induced by region equivalence (*region graph*) [2].

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

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 sections 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 $x, y \in X(P)$ such that $w(x) - w(y) > 0$: either $w'(x) - w'(y) \geq w(x) - w(y)$, or $(w'(x) - w'(y)) \approx (w(x) - w(y))$ hold.

The first condition establishes that (q, w) and (q', w') are equivalent w.r.t. all non-parametric clock constraints. The second condition ensures that, for a lower (resp. upper) bound automaton, each parametric clock constraint which is fulfilled in (q, w) (resp. (q', w')) is also fulfilled in (q', w') (resp. (q, w)). We will show that $\sqsubseteq$ indeed defines a simulation relation over the states of a lower (resp. upper) bound automaton.

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 $X(P)$;
- $c_{\mathcal{A}}$ is the maximum over $\{|c| + 1 \mid$ there is a linear expression of $\mathcal{A}$ of the form $c_0 + c_1 p_1 + \ldots + c_m p_m$ 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.

3 Emptiness and Universality for Lower Bound Automata

In this section, we study the considered decision problems for lower bound automata. We fix a lower bound automaton $\mathcal{A} = \langle Q, q^0, \Delta, F \rangle$. Also, 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 $p \in P$.

Emptiness. We recall that every linear expression of $\mathcal{A}$ is of the form $c_0 - c_1 p_1 - \ldots - c_m p_m$ with $c_i \in \mathbb{N}$ for $1 \leq i \leq m$. By decreasing the parameter values, the constraints of $\mathcal{A}$ are weakened. Thus, if $v \leq v'$ and $v' \in \Gamma(\mathcal{A})$, then also $v \in \Gamma(\mathcal{A})$ (i.e., $\Gamma(\mathcal{A})$ is *downward-closed*). Hence, to test emptiness of $\Gamma(\mathcal{A})$ it suffices to check emptiness of the *TA* $\mathcal{A}_{v_{null}}$. By Theorem 1, we obtain:

Theorem 2. *Given a lower bound automaton $\mathcal{A}$, 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 $\Gamma(\mathcal{A})$, we define a parameter valuation $v_{\mathcal{A}}$ (assigning "large" values to parameters) and show that if $v_{\mathcal{A}} \in \Gamma(\mathcal{A})$ then each $v \geq v_{\mathcal{A}}$ also belongs to $\Gamma(\mathcal{A})$. 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 timed automaton $\mathcal{A}_{v_{\mathcal{A}}}$.

Define $N_{\mathcal{A}}$ as the constant $k_{\mathcal{A}}(N_{\mathcal{R}(\mathcal{A})} + 1) + c_{\mathcal{A}}$, and denote by $v_{\mathcal{A}}$ the parameter valuation assigning $N_{\mathcal{A}}$ to each parameter. The choice of such a large constant is to ensure that in any run ρ of $[\![\mathcal{A}]\!]_{v_{\mathcal{A}}}$ we can find subruns ρ' that can be repeatedly and consecutively simulated such that we can construct a corresponding run for $[\![\mathcal{A}]\!]_v$, for any $v \geq v_{\mathcal{A}}$. Intuitively, $N_{\mathcal{A}}$ is sufficiently large to ensure that there is a portion ρ' of ρ (of duration larger than 1) which corresponds to a cycle of $\mathcal{A}_{v_{null}}$ and such that each parametric clock constraints is either always or never satisfied in all the states visited along ρ'.

A parameter valuation v *evaluates negative* for $\mathcal{A}$ if for each parametric atomic constraint $x - y \prec e$ of $\mathcal{A}$, $e[v] < 0$. Note that $v_{\mathcal{A}}$ evaluates negative for $\mathcal{A}$. We give two technical lemmas that will be used in the proof of the main theorem of this section. In these two lemmas, v is a parameter valuation which evaluates negative for $\mathcal{A}$.

Lemma 1. *[Simulation Lemma for Lower Bound Automata] Let $\rho = s_0 \xrightarrow{\delta_0,\tau_0}\!\!\!\!\!\rightarrow\; s_1 \xrightarrow{\delta_1,\tau_1}\!\!\!\!\!\rightarrow \ldots$ 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 $\rho' = s'_0 \xrightarrow{\delta_0,\tau'_0}\!\!\!\!\!\rightarrow\; s'_1 \xrightarrow{\delta_1,\tau'_1}\!\!\!\!\!\rightarrow \ldots$ such that $s'_i \sqsupseteq s_i$ for each i, and $DUR(\rho') \approx DUR(\rho)$ if ρ is finite.*

The following lemma allows us to append to a run in $[\![\mathcal{A}]\!]_v$, which corresponds to a cycle in the region graph of $\mathcal{A}_{v_{null}}$, another cycle such that its initial state s and its final state s' satisfy the strongest condition $s \sqsubseteq s'$. Note that once we apply this lemma, further cycles can be appended by repeatedly applying the Simulation Lemma. Also, note that from classical properties of timed automata the Simulation Lemma continues to hold if we replace $\sqsubseteq$ with the region equivalence $\approx_v$. However, this does not hold for the following Lemma (the properties of $\sqsubseteq$ are crucial).

Lemma 2. *Let $\rho = s_0 \xrightarrow{\delta_0,\tau_0}\!\!\!\!\!\rightarrow\; s_1 \ldots s_{n-1} \xrightarrow{\delta_{n-1},\tau_{n-1}}\!\!\!\!\!\!\!\!\rightarrow\; s_n$ be a run of $[\![\mathcal{A}]\!]_v$ such that $s_0 \approx_{v_{null}} s_n$ and for every parametric clock $x \in X(P) \setminus \{x_0\}$, if a parametric atomic constraint of the form $y - x \prec e$ appears along ρ then x is never reset along ρ. Then, there is a run $\rho' = s'_0 \xrightarrow{\delta_0,\tau'_0}\!\!\!\!\!\rightarrow\; s'_1 \ldots s'_{n-1} \xrightarrow{\delta_{n-1},\tau'_{n-1}}\!\!\!\!\!\!\!\!\rightarrow\; s'_n$ of $[\![\mathcal{A}]\!]_v$ such that $DUR(\rho') \approx DUR(\rho)$, $s'_0 = s_n$, and $s'_0 \sqsubseteq s'_n$.*

In the next theorem we show that $v_{\mathcal{A}}$ is the key valuation for reducing universality to membership to $\Gamma(\mathcal{A})$.

Theorem 3. *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 of Theorem 3: 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:

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

Observe that since $v \geq v_{\mathcal{A}}$ and $\mathcal{A}$ is a lower bound automaton, we have that v evaluates negative for $\mathcal{A}$, and in particular, $e[v] < 0$. 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 $\rho = s_0 \xrightarrow{\delta_0,\tau_0}\!\!\!\!\rightarrow\ s_1 \xrightarrow{\delta_1,\tau_1}\!\!\!\!\rightarrow\ s_2 \ldots$ 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, for $i \leq j$, denote $\rho[i,j] = s_i \xrightarrow{\delta_i,\tau_i}\!\!\!\!\rightarrow \ldots \xrightarrow{\delta_{j-1},\tau_{j-1}}\!\!\!\!\rightarrow\ s_j$.

In the rest of the proof, we first determine a finite portion of the run ρ that is crucial for the satisfaction of f_p under valuation v_p and suitable for repeated simulation, i.e., such that 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. The process is iterated until the resulting run is a run of $[\![\mathcal{A}]\!]_{v_p}$.

Assume that the clock constraint f_p appears along ρ (in the other case, ρ is also a run of $[\![\mathcal{A}]\!]_{v_p}$), and let M be the smallest index such that f_p is in the clock constraint of transition $\rho[M, M+1]$. Thus, $(v, w_M) \models f_p$. Since $f_p = z - y \prec e$ and $e[v] \leq e[v_{\mathcal{A}}] < c_{\mathcal{A}} - N_{\mathcal{A}}$, by simple arguments, it is possible to show that there are $M_y, M_z \in [0, M]$ 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 $DUR(\rho[M_y, M_z]) > N_{\mathcal{A}} - c_{\mathcal{A}}$.

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 $DUR(\rho[M_y, i]) = \tau$. 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 $DUR(\rho[i, j]) \geq 1$, $s_i \approx_{v_{null}} s_j$, and for every clock $x \in X(P) \setminus \{x_0\}$: 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 of the Claim: Let $M_y \leq K \leq M_z$ be such that $DUR(\rho[M_y, K]) = N_{\mathcal{A}} - c_{\mathcal{A}}$ (recall that $DUR(\rho[M_y, M_z]) > N_{\mathcal{A}} - c_{\mathcal{A}}$). Let $Y = \{x_1, \ldots, x_n\}$ with $n \leq k_{\mathcal{A}} - 1$ be the set of clocks in $X(P) \setminus \{x_0\}$ 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 $i_1 \leq i_2 \leq \ldots \leq i_n$. 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 $x \in X(P)$, either clock x is never reset along $\rho[i_h, i_{h+1} - 1]$ or its value is always less than $N_{\mathcal{A}} - c_{\mathcal{A}}$. Since for each parametric atomic constraint $f = x' - x \prec e'$ of $\mathcal{A}$, $e'[v] \leq e'[v_{\mathcal{A}}] < c_{\mathcal{A}} - N_{\mathcal{A}}$, we have that $(v, w) \models f$ implies $w(x) > N_{\mathcal{A}} - c_{\mathcal{A}}$. Hence, for every interval $[i_h, i_{h+1} - 1]$ and $x \in X(P) \setminus \{x_0\}$, 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 $n + 1 \leq k_{\mathcal{A}}$, $N_{\mathcal{A}} - c_{\mathcal{A}} = k_{\mathcal{A}}(N_{\mathcal{R}(\mathcal{A})} + 1)$, and $DUR(\rho[i_h - 1, i_h]) = 0$ 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 $DUR(\rho[i_k, i_{k+1} - 1]) \geq N_{\mathcal{R}(\mathcal{A})} + 1$. Recall that for each $\tau \in \mathbb{N}$ there is $i \geq M_y$ such that $DUR(\rho[M_y, i]) = \tau$, 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 $DUR(\rho[i, j]) \geq 1$ 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 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, 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. Note that 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, $DUR(\rho_h) \approx DUR(\rho[i, j])$ for $h = 1, \ldots, d$, and y is not reset in $\rho_1 \rho_2 .. \rho_d\, \rho'_{M_z}$.

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 $w(z) = 0$. Being $DUR(\rho[i, j]) \geq 1$, 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 in $[\![\mathcal{A}]\!]_{v_p}$. 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 ρ'') we get an accepting run of $[\![\mathcal{A}]\!]_{v_p}$, 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 $N_{\mathcal{A}} = O(|Q| \cdot k_{\mathcal{A}} \cdot (2c_{\mathcal{A}} + 2)^{2|X|^2})$, by Theorem 1 we obtain the following result.

Theorem 4. *Given a lower bound automaton $\mathcal{A}$, 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 Decision Problems for L/U Automata

In this section, we briefly discuss our results concerning the other decision problems we have mentioned in the introduction. We start giving the results on emptiness and universality for upper bound automata. Next, we combine the results we have given for lower bound and upper bound automata to solve such problems for general L/U automata. Then, we extend the considered problems placing linear constraints on the parameters. Finally, we use L/U automata to decide satisfiability and model-checking related problems for a dense-time linear temporal logic.

Upper bound automata. The arguments used to show the results for upper bound automata are dual to those used for lower bound automata. We fix an upper bound automaton $\mathcal{A} = \langle Q, q^0, \Delta, F\rangle$. Recall that every linear expression of $\mathcal{A}$ is of the form $c_0 + c_1p_1 + \ldots + c_mp_m$ with $c_i \in \mathbb{N}$ for each $1 \leq i \leq m$. By increasing the parameter values, the clock constraints of $\mathcal{A}$ are weakened, thus the set $\Gamma(\mathcal{A})$ is upward-closed. An immediate consequence of this property is that testing universality of $\Gamma(\mathcal{A})$ requires checking non-emptiness of the *TA* $\mathcal{A}_{v_{null}}$ (v_{null} assigns 0 to each parameter). For checking emptiness of $\Gamma(\mathcal{A})$, we establish a version of Theorem 3 for upper bound automata. Here, we use a slightly larger constant $N_{\mathcal{A}} = 8k_{\mathcal{A}}c_{\mathcal{A}}(N_{\mathcal{R}(\mathcal{A})} + 1) + c_{\mathcal{A}}$. 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 5. *Let v, v' be parameter valuations such that $v \geq v' \geq v_{\mathcal{A}}$. Then, $v \in \Gamma(\mathcal{A})$ implies $v' \in \Gamma(\mathcal{A})$.*

Since $\Gamma(\mathcal{A})$ is upward-closed, Theorem 5 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 $\mathcal{A}_{v_{\mathcal{A}}}$.

General case. 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 5, 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 $8k_{\mathcal{A}}c_{\mathcal{A}}(N_{\mathcal{R}(\mathcal{A})} + 1) + c_{\mathcal{A}}$.
- 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 $k_{\mathcal{A}}(N_{\mathcal{R}(\mathcal{A})} + 1) + c_{\mathcal{A}}$.

Thus by Theorem 1, we obtain the following result.

Theorem 6. *For an L/U automaton $\mathcal{A}$, checking for the emptiness (resp. universality) 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 $\mathcal{A}$.*

Linearly constrained parameters. A *linear constraint* C is a boolean combination of inequalities and equations of the form $e \sim 0$, where e is a linear expression and $\sim \in \{<, =\}$. 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 $Sol(C)$ we denote the set of C solutions. Given an L/U automaton $\mathcal{A}$ and a linear constraint over the $\mathcal{A}$ parameters, we consider the following decision problems:

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

We show that constrained emptiness and universality are decidable for both lower and upper bound automata. However, they become 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 separated lower bound and upper bound parameters 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 [15], and our results on unconstrained emptiness and universality.

Theorem 7. *Constrained emptiness and constrained universality are undecidable for L/U automata. However, if we restrict to 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.*

Parametric dense-time linear temporal logic. We define the logic PMITL$_{0,\infty}$ as a parametric extension of the logic MITL$_{0,\infty}$ [4]. We impose a restriction on the use of parameters reflecting that imposed on the parameters of L/U automata (by [3], if we remove this restriction, then basic decision problems become undecidable). To this aim we fix two disjoint finite sets of parameters U and L, and 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.

PMITL$_{0,\infty}$ formulas φ over a finite set AP of atomic propositions are defined as:

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

where $a \in AP$, $\prec \in \{\leq, <\}$, $\succ \in \{\geq, >\}$ and $\mathcal{U}_{\prec \mu}$ and $\mathcal{U}_{\succ \lambda}$ (resp., $\mathcal{R}_{\prec \lambda}$ and $\mathcal{R}_{\succ \mu}$) are the parameterized versions of the *until* modality (resp., *release* modality).

PMITL$_{0,\infty}$ is interpreted over *timed sequences* over 2^{AP}, defined as infinite sequences $\rho = (\sigma_0, I_0)(\sigma_1, I_1) \ldots$, where for all i, $\sigma_i \in 2^{AP}$, and $I_0, I_1, \ldots$ represents a partition of $\mathbb{R}_{\geq 0}$ in non-empty intervals such that for all i, the upper bound of I_i equals the lower bound of I_{i+1}. For $t \in \mathbb{R}_{\geq 0}$, let $\rho(t)$ be the unique σ_i such that $t \in I_i$.

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

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

For a formula φ, a timed sequence ρ, and a parameter valuation v, ρ *satisfies* φ *under valuation* v if $(\rho, v, 0) \models \varphi$. Note that we have defined PMITL$_{0,\infty}$ formulas in positive normal form. It is simple to verify that the until and the release operators are dual, and therefore, the logic is closed under semantic negation.

For such a logic, we study the related satisfiability and model-checking problems. For a given PMITL$_{0,\infty}$ formula φ and an L/U automaton $\mathcal{A}$ such that the lower (resp., upper) bound parameters of $\mathcal{A}$ are from L (resp., U), we consider the emptiness and universality problems for the following sets of parameter valuations: the set $S(\varphi)$ of parameter valuations that make φ satisfiable, and the set $V(\mathcal{A}, \varphi)$ of parameter valuations v for which every timed sequence accepted by $[\![\mathcal{A}]\!]_v$ satisfies φ. Note that the semantics of L/U automata can be slightly modified such that an L/U automaton recognizes timed sequences (see [4] for standard timed automata).

We solve the above decision problems by reducing them to corresponding problems on L/U automata. The key of these reductions is the translation of a PMITL$_{0,\infty}$ formula into an equivalent L/U automaton. Such translation relies on the construction given in [4] for MITL$_{0,\infty}$ and *TA*.

Theorem 8. *For a PMITL$_{0,\infty}$ formula φ and an L/U automaton $\mathcal{A}$, checking for emptiness and universality of $S(\varphi)$ and $V(\mathcal{A}, \varphi)$ is* PSPACE*-complete.*

5 Conclusion

We have studied some decision problems on L/U automata. In particular, we have shown that the emptiness and universality problems for the set of parameter valuations for which there is an infinite accepting run are decidable and PSPACE-complete. This allows us to prove decidability of a parametric extension of MITL$_{0,\infty}$. Furthermore, we have studied a constrained version of emptiness and universality with parameters constrained by linear systems of equations and inequalities. 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.

There are other results that can be derived from those presented here. As an example, we could combine the results on constrained decision problems along with those on PMITL$_{0,\infty}$ to solve the constrained versions of the decision problems for PMITL$_{0,\infty}$. Moreover, 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 constraints over parameters (this can be done similarly to what is done

in [3] for PLTL). Also, we can solve some optimization problems on the parameter valuations, which can be very interesting for system designers, and decide the finiteness of the set $\Gamma(\mathcal{A})$.

As future research, we think of the extension of our results to real-valued parameters. The results we have shown in this paper answer only partially to this problem. Another interesting direction is to investigate the parametric extension of MITL [4], where constraints are expressed in form of intervals as opposed to bounds as in $\text{PMITL}_{0,\infty}$. The technique we have used here for $\text{PMITL}_{0,\infty}$ does not seem to scale to such a logic, and a different approach may be required.

References

1. Alur, R., Courcoubetis, C., Dill, D.L.: Model-checking in dense real-time. Information and Computation 104(1), 2–34 (1993)
2. Alur, R., Dill, D.: A theory of timed automata. Theoretical Computer Science 126(2), 183–235 (1994)
3. Alur, R., Etessami, K., La Torre, S., Peled, D.: Parametric temporal logic for model measuring. ACM Transactions on Computational Logic 2(3), 388–407 (2001)
4. Alur, R., Feder, T., Henzinger, Th.A.: The benefits of relaxing punctuality. Journal of the ACM 43(1), 116–146 (1996)
5. Alur, R., Henzinger, Th.A., Vardi, M.Y.: Parametric real-time reasoning. In: Proc. of the 25th ACM Symposium on Theory of Computing (STOC'93), pp. 592–601. ACM Press, New York (1993)
6. Alur, R., Madhusudan, P.: Decision problems for timed automata: a survey. In: Bernardo, M., Corradini, F. (eds.) Formal Methods for the Design of Real-Time Systems. LNCS, vol. 3185, pp. 1–24. Springer, Heidelberg (2004)
7. Bruyère, V., Dall'Olio, E., Raskin, J.F.: Durations, Parametric Model-Checking in Timed Automata with Presburger Arithmetic. In: Alt, H., Habib, M. (eds.) STACS 2003. LNCS, vol. 2607, pp. 687–698. Springer, Heidelberg (2003)
8. Bruyère, V., Raskin, J.F.: Real-time model-checking: Parameters everywhere. In: Pandya, P.K., Radhakrishnan, J. (eds.) FST TCS 2003: Foundations of Software Technology and Theoretical Computer Science. LNCS, vol. 2914, pp. 100–111. Springer, Heidelberg (2003)
9. Emerson, E.A., Trefler, R.: Parametric Quantitative Temporal Reasoning. In: Proc. 14th Ann. Symp. Logic in Computer Science (LICS'99), pp. 336–343. IEEE Computer Society Press, Los Alamitos (1999)
10. Henzinger, T., Kopke, P., Puri, A., Varaiya, P.: What's decidable about hybrid automata. Journal of Computer and System Sciences 57, 94–124 (1998)
11. Hune, T., Romijn, J., Stoelinga, M., Vaandrager, F.: Linear parametric model checking of timed automata. Journal of Logic and Algebraic Programming 52,53, 183–220 (2002)
12. IEEE Computer Society. IEEE Standard for a High Performance Serial Bus. Std 1394-1995 (August 1996)
13. Lamport, L.: A Fast Mutual Exclusion Algorithm. ACM Transactions Computer Systems 5(1), 1–11 (1987)
14. Pnueli, A.: The temporal logic of programs. In: Proc. of the 18th IEEE Symposium on Foundations of Computer Science, pp. 46–77. IEEE Computer Society Press, Los Alamitos (1977)
15. Pottier, L.: Minimal solutions of linear diophantine systems: bounds and algorithms. In: Book, R.V. (ed.) Rewriting Techniques and Applications. LNCS, vol. 488, pp. 162–173. Springer, Heidelberg (1991)
16. Wang, F.: Parametric timing analysis for real-time systems. Information and Computation 130(2), 131–150 (1996)