Theoretical Computer Science 564 (2015) 131–148

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Parametric metric interval temporal logic ☆

Barbara Di Giampaolo, Salvatore La Torre, Margherita Napoli *

Dipartimento di Informatica, Università di Salerno, Italy

ARTICLE INFO

Article history:
Received 9 September 2013
Received in revised form 4 October 2014
Accepted 17 November 2014
Available online 22 November 2014
Communicated by D. Sannella

Keywords:
Temporal logic
Parametric real-time verification
(Parametric) timed automata

ABSTRACT

In this paper we focus on the role of parametric constants in real-time temporal logic and introduce the logic PMITL as a parametric extension of MITL. For this logic, we study decision problems which are the analogues of satisfiability, validity and model-checking problems for non-parametric temporal logic. We impose some restrictions on the use of the parameters: each parameter is used with a fixed polarity, parameters can appear only in one of the endpoints of the intervals, parametric linear expressions can be used only as right endpoints of the intervals. We show that, for parameter valuations yielding only non-singular intervals, the considered problems are all decidable and EXPSPACE-complete, such as for the decision problems in MITL. Moreover, we show that if we relax any of the imposed restrictions, the problems become undecidable. We also investigate the computational complexity of these problems for natural fragments of PMITL, and show that for some meaningful fragments they can be solved in polynomial space and are PSPACE-complete. Finally, we consider the decision problem of determining the truth of first-order queries over PMITL formulas where the parameters are used as variables that can be existentially or universally quantified. We solve this problem in several cases and exhibit an exponential-space algorithm.

1. Introduction

Temporal logic is a simple and standard formalism to specify the desired behavior of a reactive system. Its use as a specification language was first suggested by Pnueli [24] who proposed the propositional linear temporal logic (LTL). This logic presents natural operators to express temporal requests on the time ordering of occurrences of events, such as “always”, “eventually”, “until”, and “next”.

The logics MTL [20] and MITL [3] extend LTL with a real-time semantics where the changes of truth values happen according to a splitting of the line of non-negative reals into intervals. Syntactically, these logics augment the temporal operators of LTL (except for the next operator which has no clear meaning in a real-time semantics) with a subscript which expresses an interval of interest for the expressed property. Thus, properties such as “every time an a occurs then a b must occur within time $t \in [3, 5]$” become expressible. To gain the decidability, the singular intervals, or equivalently the equality, are not allowed as subscripts of the temporal operators in MITL [3].

In this paper, we extend MITL with parametric constants, i.e., we allow the intervals in the subscripts of the temporal operators to have as an endpoint a *parametric expression* of the form $c + x$, for a parameter x and a constant c. Therefore,

☆ This work was partially funded by the MIUR grants FARB 2012–2014 Università degli Studi di Salerno (Italy).
* Corresponding author.
E-mail addresses: slatorre@unisa.it (S. La Torre), napoli@unisa.it (M. Napoli).

http://dx.doi.org/10.1016/j.tcs.2014.11.019
0304-3975/

typical time properties which are expressible in MITL can now be analyzed by varying the scope of the temporal operators depending on the values of the parameters. As an example, consider a parameterized version of the above property: "every time an a occurs then a b must occur within a time $t \in [3, 5 + x]$" where x is a parameter. One could be interested in determining if there exists a value of x such that this property holds on a given timed sequence, or if this is true for any possible value of x, or more, the set of x values such that the property holds.

We define a logic, which we denote PMITL (parametric MITL), where in each interval at most one endpoint can be a parametric expression, and such that each parameter is used with a fixed "polarity". The concept of polarity is semantic and is related to whether the space of the values for a parameter such that the formula is satisfied is upward or downward closed. For example, the set of values of x for the assertion "an a will eventually occur within time x" is upward closed: a model which satisfies this for $x = 3$ also satisfies it for every $x > 3$.

For the introduced logic, we study the decision problems which are the analogues of satisfiability, validity and model-checking for non-parametric temporal logic. As models for our formulas, we consider the class of timed automata augmented with parameters, which are called L/U automata [17] and share the same kind of restrictions on the parameters as PMITL. Note that in the model-checking problem we allow parameters both in the formula and the model.

We define our decision problems as the universality and the emptiness problems for classes of sets of parameter valuations. In particular, we define the set of all the admissible valuations such that a formula is satisfiable and the set of all the admissible valuations such that an L/U automaton is a model of the formula (in the following, referred to as S-sets), and the analogous sets which guarantee instead the validity of a given formula, possibly with respect to a given L/U automaton (in the following, referred to as V-sets). Here, a parameter valuation is admissible if it evaluates a parameterized interval to neither an empty nor a singular set. As in MITL, the restriction to non-singular intervals is required for the decidability of the decision problems. The other restriction is not critical for decidability, and has the only purpose to rule out some degenerate cases: an empty interval would make the related assertion equivalent to either the constant true or false.

Our main result is that the universality and the emptiness problems for the S-sets and the V-sets are decidable and Expspace-complete. The result still holds if we allow linear expressions of the parameters, of the form $c_1x_1 + \ldots + c_nx_n$, provided that such expressions are used only as right endpoints of the parameterized intervals. The proof goes through a reduction to decidable problems for Büchi L/U automata [7]. Recall that satisfiability, validity and model-checking problems for MITL formulas [3] are Expspace-hard, thus this result shows that adding parameters to MITL, though augmenting the expressiveness, does not affect the abstract computational complexity of its decision problems.

One of the crucial points concerning parametric systems is to explicitly compute the set of all the valuations that make a parameterized formula satisfiable (or valid). In [2] this problem has been solved for a parametric logic (PLTL) with a discrete semantics when all the operators have the same polarity, using an algorithm that takes double-exponential time in the number of parameters. It is very interesting to seek algorithms which get more information on this set of parameter valuations also when parameters of both polarities are allowed. We make a progress in this direction by studying a general decision problem over the S-sets and V-sets (that includes the above emptiness and universality problems as particular cases).

For a given formula with parameters $x_1, \ldots, x_n$, we study queries of the form $Q_1x_1 \ldots Q_nx_n$. over the corresponding S-sets and V-sets, where each Q_i is either an existential or a universal quantifier. For each i, the quantifier Q_i bounds, either existentially or universally, the values that a parameter x_i may be assigned within the given set. The already discussed non-emptiness problem corresponds to queries with only existential quantifiers, and the universality problem corresponds to queries with only universal quantifiers. We prove that also this generalized decision problem is not harder than the basic problems and that this holds not only when all the parameters have the same polarity but also in several more general cases.

We refine our complexity results by addressing the complexity of some fragments of PMITL. On the positive side, we prove that some of the considered problems are in Pspace for two expressive fragments. In particular, we prove that the following problems are in Pspace: (1) deciding the emptiness of S-sets for formulas where the only parameterized operators are either of the form $\diamond_{(c,d+x)}$, or one of the endpoints of its interval is 0 or ∞, (2) deciding the universality of V-sets in the fragment where the only parameterized operators are either of the form $\square_{(c,d+y)}$, or one of the interval endpoints is 0 or ∞. These fragments are quite expressive (for example we can express properties such as the parameterized response property seen above) and contain the logic considered in [7]. To the best of our knowledge, the union of these fragments captures the most general known formulation of parametric constraints in PMITL with the considered decision problems in Pspace.

We complete our analysis on the complexity of the fragments by showing that the considered decision problems are Expspace-hard for the fragment $\text{PMITL}_{0,\infty}$ where each non-parametric interval has one endpoint which is either 0 or ∞, and that hardness still holds in the fragments of $\text{PMITL}_{0,\infty}$ with only parametric operators of one polarity.

In the definition of PMITL we impose some restrictions on the parameters. In particular, we do not allow a parameter valuation to evaluate a parametric interval to a singular interval, each parameter is used with a fixed polarity consistently through all the formula, and in each interval we can use parameters in only one of the endpoints. We show that if we relax any of these restrictions the resulting problems become undecidable. In fact, it is known that testing parameters for equality (and thus allowing singular intervals) leads to undecidability (see [2]). Also, if a parameter is used with both polarities in a formula, then it is possible to express equality, and again the problems become undecidable. We thus consider some natural ways of defining parameterized intervals with parameters in both the endpoints (i.e., both the endpoints are added with the same parameter or with different parameters) and finally we allow parametric linear expressions as left endpoint of the parameterized intervals. In all such cases, the studied decision problems become undecidable in the resulting logic.

The rest of the paper is organized as follows. Section 1.1 discusses some related work. Section 2 contains the basic definitions. Section 3 introduces the logic PMITL along with the related decision problems, and gives a comparison with MITL. In Section 4, some preliminary results are shown which give further insights on the logic. In Section 5, we prove the decidability results of the satisfiability, validity, and model-checking problems. Section 6 is dedicated to the study of the computational complexity of natural syntactic fragments of PMITL. Section 7 deals with generalizations of our decidability results. In Section 8, we show that relaxing the restrictions imposed over the parameterized intervals leads to undecidability. Finally, in Section 9, we conclude with some remarks and future directions.

1.1. Related work

A very preliminary version of this paper is [14]. The use of parametric constants has been advocated by many authors as a support to designers in the early stages of a design when not much is known on the system under construction (see for example [4,2,9,13,11,15,17,25]). The need for restricting the use of parameters (in order to obtain decidability) such that each parameter is always used with a fixed polarity was addressed already in [2] for obtaining decidability results for a parametric extension of LTL (denoted PLTL). In [21], the results on PLTL have been shown using different techniques.

A parameterized fragment of PMITL is studied also in [7]. Both in [2] and [7], time constraints on temporal operators do not allow intervals with arbitrary endpoints: one of the endpoints is always 0 or ∞. Here, we give a thorough study of the parameterization with intervals and get a deeper insight on some concepts expressed there. The techniques used to show decidability results in [2,7], cannot be used directly in our settings, and we would like to stress that in the beginning it was not clear to us that PMITL was even decidable.

Parametric branching time specifications were first investigated in [25,15] where decidability is shown for logics obtained as extensions of TCTL [1] with parameters. In [8], decidability is extended to full TCTL with Presburger constraints over parameters. In [9], decidability is established for the model checking problem of *discrete-time* timed automata with *one parametric clock* against parametric TCTL without equality.

Parametric timed automata were introduced in [4], where the reachability problem is shown to be undecidable for automata with three or more parameterized clocks, and decidable when only one parameteri zed clock is used. A recent result has shown that this problem is also decidable when only two parameterized clocks and one parameter are used [10].

L/U automata were introduced in [17] as a restriction of parametric timed automata where each parameter occurs either as a lower bound or as an upper bound in the timing constraints. The emptiness problem for this model is shown to be decidable with respect to finite runs in [17] and to a Büchi acceptance condition in [7]. In [7], also the parameter synthesis problem is addressed when only parameters of the same polarity are used. In [18], the undecidability results on parametric timed automata are further refined showing that the unavoidability problem (i.e., all runs must go through a target set) for parametric timed automata where parameters occur only as upper bounds in the timing constraints is undecidable. The authors also propose and study a restriction that limits the search for parameter values to bounded integers. A parameter synthesis algorithm for parametric timed automata that starts from a reference parameter valuation and derives constraints on parameters preserving time-abstract equivalence is given in [5]. The synthesis of parameters for temporal logic over signals are addressed in [6], where both magnitude and timing parameters are considered, and in [26], where only one parameter is allowed and the synthesis is achieved via an optimization problem.

2. Preliminaries

Notation. A *time interval* is any interval of non-negative real numbers. In the following, when we need to stress only the endpoints of an interval, we use (a,b) to denote any interval with a left endpoint a and a right endpoint b. To denote specific intervals, we use the notation $[a,b]$, $]a,b[$, $[a,b[$, and $]a,b]$ respectively for the closed, open, left-closed/right-open and left-open/right-closed interval. When $a > b$ (or $a \geq b$, if the interval is not closed) (a,b) is the empty interval. A closed interval $[a,a]$ is called *singular*. Given an interval $I = (a,b)$ and $t \geq -a$, with $I + t$ we denote the interval $(a+t, b+t)$ such that I is left-closed (resp. right-closed) iff $I + t$ is left-closed (resp. right-closed).

An *interval sequence* is an infinite sequence $I_0, I_1 \ldots$ of non-empty time intervals such that:

- for all i, $I_i \cap I_{i+1} = \emptyset$ and, denoting $I_i = (a_i, b_i)$, $a_{i+1} = b_i$ holds (*along the time line I_{i+1} follows I_i*);
- each real number $t \geq 0$ belongs to some interval I_i (*the sequence of intervals covers the reals*).

In the rest of the paper, we fix a set of atomic propositions AP. A *timed sequence* over AP is an infinite sequence $\alpha = \langle \alpha_0, I_0 \rangle \langle \alpha_1, I_1 \rangle \ldots$ such that $\alpha_i \in 2^{AP}$, for all i, and $I_0, I_1 \ldots$ is an interval sequence. For each $t \geq 0$, $\alpha(t)$ denotes the unique α_i such that $t \in I_i$. Two timed sequences α' and α'' are equivalent if $\alpha'(t) = \alpha''(t)$ for all $t \geq 0$.

In the rest of the paper U and L denote two disjoint sets of parameters. A *parametric expression* is an expression of the form $c+z$, where $c \in \mathbb{N}$ and z is a parameter. With $\mathcal{E}(U)$ (resp. $\mathcal{E}(L)$) we denote the set of all the parametric expressions over parameters from U (resp. L). A *parameterized interval* is a time interval (a,b) such that either a or b belong to $\mathcal{E}(L) \cup \mathcal{E}(U)$

and the other one belongs to $\mathbb{N}$. In the following, we sometimes use the term interval to indicate either a parameterized interval or a time interval. A *parameter valuation* $v : L \cup U \longrightarrow \mathbb{N}$ assigns a natural number to each parameter. Given a parameter valuation v and an interval I, with I_v we denote the time interval obtained by evaluating the endpoints of I by v (in particular, if I is a time interval then $I_v = I$). For parameter valuations v, v' and $\approx \in \{<, \leq, =, \geq, >\}$, we use the standard notation $v \approx v'$ to denote the extension of $\approx$ to tuples, that is, the relation defined as: $v(z) \approx v'(z)$ for each parameter z.

Parametric Timed Automata. *Parametric Timed Automata* extend Timed Automata, allowing the use of parameters in the clock constraints, see [4]. We briefly recall the definition. Given a finite set of clocks X and a finite set of parameters P, a *parametric clock constraint* is a positive boolean combination of terms of the form $\xi \approx e$ where $\xi \in X$, $\approx \in \{<, \leq, \geq, >\}$, and either $e \in \mathbb{N}$ or e is a parametric expression with variables in P. Let Ξ denote the set of parametric clock constraints over X and P. A *parametric timed automaton* (PTA) is a tuple $\mathcal{A} = (Q, Q^0, X, P, \beta, \Delta, \lambda)$, where Q is a finite set of *locations*, $Q^0 \subseteq Q$ is the set of *initial locations*, X is a finite set of clocks, P is a finite set of parameters, $\beta : Q \to \Xi$ is a function assigning to each location an *invariant* (parametric clock constraint over X and P), $\Delta \subseteq Q \times \Xi \times Q \times 2^X$ is a *transition relation*, and $\lambda : Q \to 2^{AP}$ is a function labeling each location with a set of atomic propositions. A *timed automaton* is a PTA where all the parametric clock constraints are just clocks compared to constants in $\mathbb{N}$.

A *clock interpretation* $\sigma : X \to \mathbb{R}_+$ assigns real values to each clock. A clock interpretation $\sigma + t$, for $t \in \mathbb{R}_+$, assigns $\sigma(\xi) + t$ to each $\xi \in X$. For $\gamma \subseteq X$, $\sigma[\gamma := 0]$ denotes the clock interpretation that assigns 0 to all clocks in γ, and $\sigma(\xi)$ to all the other clocks ξ. We say that a clock interpretation σ and a parameter valuation v satisfy a parametric clock constraint δ, denoted $(\sigma, v) \models \delta$, if evaluating each clock of δ according to σ and each parameter of δ according to v, the resulting boolean expression holds true.

For locations $q_i \in Q$, clock interpretations σ_i, clock constraints $\delta_i \in \Xi$, clock sets $\gamma_i \subseteq X$, and intervals I_i, a *run* ρ of a PTA $\mathcal{A}$, under a parameter valuation v, is an infinite sequence $\xrightarrow{\sigma_0} \langle q_0, I_0 \rangle \xrightarrow[\delta_1, \gamma_1]{\sigma_1} \langle q_1, I_1 \rangle \xrightarrow[\delta_2, \gamma_2]{\sigma_2} \langle q_2, I_2 \rangle \xrightarrow[\delta_3, \gamma_3]{\sigma_3} \ldots$, such that:

- $q_0 \in Q^0$, and $\sigma_0(\xi) = 0$, for each clock ξ;
- $I_0, I_1 \ldots$ is an interval sequence;
- for all $i \geq 0$, denoting $I_i = (a_i, b_i)$ and $\sigma_i' = \sigma_i + b_i - a_i$:
  (1) $(q_i, \delta_{i+1}, q_{i+1}, \gamma_{i+1}) \in \Delta$, $(\sigma_i', v) \models \delta_{i+1}$ and $\sigma_{i+1} = \sigma_i'[\gamma_{i+1} := 0]$;
  (2) $\forall t \in I_i$, $\sigma_i + (t - a_i)$ and v satisfy $\beta(q_i)$.

The timed sequence associated with ρ is $\langle \lambda(q_0), I_0 \rangle \langle \lambda(q_1), I_1 \rangle \langle \lambda(q_2), I_2 \rangle \ldots$. With $L(\mathcal{A}, v)$ we denote the set of the timed sequences over AP which are equivalent to those associated with a run of $\mathcal{A}$ under a parameter valuation v.

We recall that for parametric timed automata the reachability problem, stated as determining the existence of a parameter valuation for which a target set of locations is reachable, is in general undecidable [4]. An interesting decidable subclass of PTA is that of *lower bound/upper bound* (L/U) *automata* which are defined as PTA such that the set L of the parameters which can occur as a lower bound in a parametric clock constraint is disjoint from set U of the parameters which can occur as an upper bound. The intuition behind the decidability of this model is that decision problems can be reduced to queries on timed automata for bounded parameter valuations (see [7]). A similar property was observed also for parametric discrete-time linear temporal logic [2].

For L/U automata both acceptance criteria on finite runs and on infinite runs have been considered [17,7]. Here, we recall the Büchi acceptance condition. A *Büchi* L/U automaton $\mathcal{A}$ is an L/U automaton coupled with a subset F of locations. A run ρ is accepting for $\mathcal{A}$ if at least one location in F repeats infinitely often along ρ. We denote with $\Gamma(\mathcal{A})$ the set of parameter valuations v such that there exists an accepting run under v.

We recall the following result.

Theorem 1. *(See [7].) The problems of checking the emptiness and the universality of $\Gamma(\mathcal{A})$, for a Büchi L/U automaton $\mathcal{A}$, are* PSPACE*-complete.*

Example 1. Consider a wire component in a synchronous circuit. This is a very simple timed system that carries an input signal to its output with a delay belonging to $[l^\uparrow, u^\uparrow]$ or $[l^\downarrow, u^\downarrow]$ depending on whether the input signal is propagated on a rising (from low to high) or a falling (from high to low) edge. (Observe that in real circuits, the propagation interval of the rising and falling edges differ.) The timed automaton in Fig. 1(a) is a model for a wire component given in [12] as a part of a model for the SPSMALL memory, a commercial product of STMicroeletronics. There, the entire memory is modeled as the synchronous product of the timed automata corresponding to the input signals and the internal components of the memory, such as latches, wires and logical blocks.

The timed automaton in Fig. 1(a) has one clock variable z_0 and five locations. The symbol $r^\uparrow$ (resp. $r^\downarrow$) labels the location which is entered when the input signal r is rising (resp. falling), and similarly $o^\uparrow$ (resp. $o^\downarrow$) for the output signal o. Each edge corresponds to a discrete event in the system. The location labeled with $r^\uparrow$ (resp. $r^\downarrow$) is associated with invariant $z_0 \leq u^\uparrow$ (resp. $z_0 \leq u^\downarrow$), and the guard associated with the outgoing transition to the location labeled with $o^\uparrow$ (resp. $o^\downarrow$) is labeled with $z_0 \geq l^\uparrow$ (resp. $z_0 \geq l^\downarrow$).

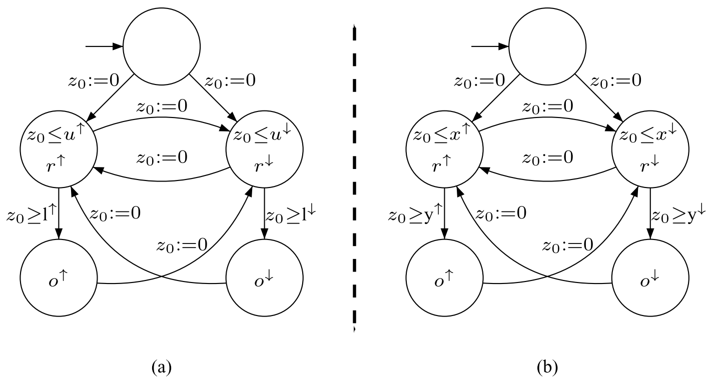

Fig. 1. Models of a wire component: (a) timed automaton, (b) L/U automaton.

When the data-sheet of a circuit does not provide enough information, we can use parameters to formulate the wished system requirements and then perform a parameterized analysis of the circuit (that essentially asks to solving the introduced decision problems). In particular, for the wire model we replace the constants $l^{\uparrow}$, $l^{\downarrow}$, $u^{\uparrow}$, $u^{\downarrow}$ with the parameters $x^{\uparrow}$, $x^{\downarrow}$, $y^{\uparrow}$, $y^{\downarrow}$, respectively. The resulting parametric timed automaton is given in Fig. 1(b). We observe that $x^{\uparrow}$ and $x^{\downarrow}$ are only used as upper bounds in the parametric clock constraints. Similarly, $y^{\uparrow}$ and $y^{\downarrow}$ are only used as lower bounds. Therefore, the PTA from Fig. 1(b) is an L/U automaton. $\square$

3. Parametric dense-time Metric Interval Temporal Logic

Syntax. The *Parametric dense-time Metric Interval Temporal Logic* (PMITL) formulas over *AP* are defined by the following grammar:

$$\varphi := p \mid \neg p \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \varphi \mathcal{U}_H \varphi \mid \varphi \mathcal{R}_J \varphi,$$

where $p \in AP$, and H and J are either non-empty and non-singular time intervals with endpoints in $\mathbb{N} \cup \{\infty\}$ or parameterized time intervals such that:

- for $H = (a, b)$ either (1) $a \in \mathbb{N}$ and $b \in \mathcal{E}(U)$ or (2) $a \in \mathcal{E}(L)$ and $b \in (\mathbb{N} \cup \{\infty\})$;
- for $J = (a, b)$ either (1) $a \in \mathbb{N}$ and $b \in \mathcal{E}(L)$ or (2) $a \in \mathcal{E}(U)$ and $b \in (\mathbb{N} \cup \{\infty\})$.

Observe that only one of the two endpoints of any interval used in the subscripts may be parametric.

As usual, we use the abbreviations $\Diamond_H \varphi$ and $\Box_J \varphi$ for *true* $\mathcal{U}_H \varphi$ and *false* $\mathcal{R}_J \varphi$, respectively.

Recall that MITL [3] is defined as above where subscripts H and J are non-singular time intervals with endpoints in $\mathbb{N} \cup \{\infty\}$.

The logic $\text{MITL}_{0,\infty}$ is defined in [3] as a syntactic restriction of MITL where all the time intervals (a, b), that are used as subscripts of the temporal operators, are such that either $a = 0$ or $b = \infty$. With $\text{PMITL}_{0,\infty}$ we denote the parametric extension of $\text{MITL}_{0,\infty}$ which corresponds to the fragment of PMITL where all the time (non-parameterized) intervals are as in $\text{MITL}_{0,\infty}$. Note that in [7], the acronym $\text{PMITL}_{0,\infty}$ is used to denote the parametric extension of $\text{MITL}_{0,\infty}$ where also the parameterized intervals are restricted such that one of the endpoints is either 0 or ∞. Here, we prefer to denote this extension of $\text{MITL}_{0,\infty}$ as $\text{P}_{0,\infty}\text{MITL}_{0,\infty}$ to stress the fact that the imposed syntactic restriction concerns both time and parametric intervals.

We consider also two interesting fragments of $\text{PMITL}_{0,\infty}$: $\text{PMITL}_{\Diamond}$ is the fragment where the only parameterized operators are either of the form $\Diamond_{(c,d+x)}$, for $x \in U$, or one of the interval endpoints is 0 or ∞, and $\text{PMITL}_{\Box}$ is the fragment of $\text{PMITL}_{0,\infty}$ where the only parameterized operators are either of the form $\Box_{(c,d+y)}$, for $y \in L$, or one of the interval endpoints is 0 or ∞.

All the considered fragments of PMITL are reported in Fig. 2, where the edges denote syntactic inclusion between the connected classes, from the top to the bottom.

In the following, given a formula φ we denote with K_φ the maximal constant used in φ augmented by 1 and with N_φ the number of subformulas of φ.

Semantics. PMITL formulas are interpreted over timed sequences and with respect to a parameter valuation. For a formula φ, a timed sequence α, a parameter valuation v, and $t \in \mathbb{R}_+$, the *satisfaction relation under valuation* v, denoted $(\alpha, v, t) \models \varphi$, is defined as follows:

- $(\alpha, v, t) \models p \Leftrightarrow p \in \alpha(t)$;

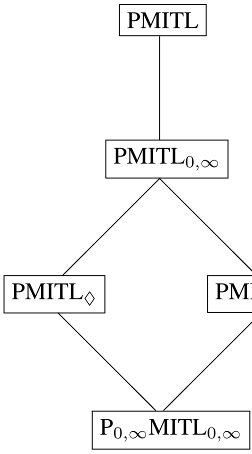

Fig. 2. The hierarchy of PMITL fragments with respect to syntactic inclusion.

- $(\alpha, v, t) \models \neg p \Leftrightarrow p \notin \alpha(t)$;
- $(\alpha, v, t) \models \varphi \wedge \psi \Leftrightarrow$ both $(\alpha, v, t) \models \varphi$ and $(\alpha, v, t) \models \psi$;
- $(\alpha, v, t) \models \varphi \vee \psi \Leftrightarrow (\alpha, v, t) \models \varphi$ or $(\alpha, v, t) \models \psi$;
- $(\alpha, v, t) \models \varphi \mathcal{U}_H \psi \Leftrightarrow$ for some $t' \in H_v + t$, $(\alpha, v, t') \models \psi$, and $(\alpha, v, t'') \models \varphi$ for all $t < t'' < t'$;
- $(\alpha, v, t) \models \varphi \mathcal{R}_J \psi \Leftrightarrow$ either $(\alpha, v, t') \models \psi$, for all $t' \in J_v + t$, or there exists $t'' > t$ such that $(\alpha, v, t'') \models \varphi$ and $(\alpha, v, t') \models \psi$ for all $t' \in [t, t''] \cap J_v + t$.

A timed sequence α *satisfies* φ *under valuation* v, denoted $(\alpha, v) \models \varphi$, if $(\alpha, v, 0) \models \varphi$. Observe that the temporal operators $\mathcal{U}$ and $\mathcal{R}$ are dual.

PMITL formulas can express meaningful properties of real-time systems. Sample PMITL formulas are $\varphi_1 = \Box(r^{\uparrow} \Rightarrow \Diamond_{[c,d+x]} o^{\uparrow})$ and $\varphi_2 = \Box(r^{\downarrow} \Rightarrow \Diamond_{[c+y,d]} o^{\downarrow})$, where $c, d \in \mathbb{N}$, $x \in U$, $y \in L$, and the atomic propositions are from Example 1. These formulas are variations with parameters of the standard time response property. In particular, φ_1 asserts that "every rising edge of the input signal r is followed by a rising edge of the output signal o within the interval $[c, d+x]$". In the first formula (which is a $\mathrm{PMITL}_{\Diamond}$ formula), we fix a constant lower bound on the response property and restrict the upper bound to be at least d. In the second formula instead, we fix an upper bound d and require the lower bound to be at least c.

PMITL vs. MITL. The logic PMITL adds to MITL in terms of expressiveness, in the sense that some properties can be expressed in PMITL but not in MITL. More precisely, some sets of timed sequences can be characterized by a PMITL formula but they cannot be defined by any MITL formula. Given a PMITL formula φ, denote with $L_{exist}(\varphi)$ the set of timed sequences $\{\alpha | \exists v$ such that $(\alpha, v) \models \varphi\}$. The following proposition holds.

Proposition 1. *Let φ be the PMITL formula $\Box(\neg p \Rightarrow \Diamond_{]0,x]} p)$, for $x \in U$ and an atomic proposition p. There is no MITL formula φ' such that the set of the timed sequences satisfying φ' is equal $L_{exist}(\varphi)$.*

Proof. Observe that the set $L_{exist}(\varphi)$ contains all the timed sequences α such that there is a bound $b_\alpha \in \mathbb{N}$ on the length of all the intervals where $\neg p$ holds. Now, let φ' be an MITL formula which is satisfied by all the timed sequences of $L_{exist}(\varphi)$ (if such a formula does not exist we are done). We will show that there is a timed sequence $\beta \notin L_{exist}(\varphi)$ but satisfying φ'. For $m \in \mathbb{N}$, let α_m be the timed sequence such that the proposition p holds only at all the times nm, for $n \in \mathbb{N}$. Clearly, $\alpha_m \in L_{exist}(\varphi)$.

For an MITL formula ψ, denote with *Closure*(ψ) the set containing all the subformulas of ψ and, for each subformula of the form $\psi' O_{(c,d)} \psi''$, for O being either $\mathcal{U}$ or $\mathcal{R}$, all the formulas $\psi' O_{(c-b,d-b)} \psi''$, for a non-negative $b \leq c$, and all the formulas $\psi' O_{(0,a)} \psi''$, for $0 < a < d - c$. Given a timed sequence α, we can label each time t on α with all the formulas of *Closure*(ψ) which are fulfilled by the suffix of α starting at t. We call a *cycle* with respect to ψ of α any bounded portion of α from a time t to time t' such that $t' - t > 1$, and t and t' are labeled with the same set of formulas from *Closure*(ψ). Denoting with α' a timed sequence obtained from α by iterating an arbitrary number of times a cycle with respect to ψ, by induction on the structure of MITL formulas, one can show that if α satisfies ψ, then also α' satisfies ψ.

Now, take $m > 2^{N_{\varphi'} K_{\varphi'}}$. Since the number of formulas in *Closure*(φ') is bounded by $N_{\varphi'} K_{\varphi'}$, we are guaranteed that in each portion of α_m between two consecutive occurrences of p, there is a cycle with respect to φ'. Therefore, we can iterate such cycles such that in the resulting sequence β the distance between two consecutive occurrences of p grows unboundedly. Clearly, β does not belong to $L_{exist}(\varphi)$ but fulfills φ'. Therefore, there are no MITL formulas that can express φ. $\Box$

Decision problems. Let us first introduce a further restriction on the parameter valuation that is crucial for decidability.

For a given formula φ, a parameter valuation v is *admissible* for φ if for each interval I in the subscripts of φ, I_v is either a non-singular time interval or the interval $[0, 0]$. With $\mathcal{D}(\varphi)$ we denote the set of all the admissible valuations for φ.

Two PMITL formulas φ_1 and φ_2 are *equivalent* if $\mathcal{D}(\varphi_1) = \mathcal{D}(\varphi_2)$ and for all the timed sequences α and all the parameter valuations $v \in \mathcal{D}(\varphi_1)$, $(\alpha, v) \models \varphi_1$ if and only if $(\alpha, v) \models \varphi_2$.

For the logic PMITL, we study the analogous problems to satisfiability, validity, and model-checking for standard temporal logic. More precisely, given a PMITL formula φ and an L/U automaton $\mathcal{A}$, consider the following sets:

- the set $S(\varphi)$ of the parameter valuations $v \in \mathcal{D}(\varphi)$ such that there is a timed sequence that satisfies φ under valuation v;
- the set $V(\varphi)$ of the parameter valuations $v \in \mathcal{D}(\varphi)$ such that all the timed sequences satisfy φ under valuation v;
- the set $S^*(\mathcal{A}, \varphi)$ of the parameter valuations $v \in \mathcal{D}(\varphi)$ such that there is a timed sequence in $L(\mathcal{A}, v)$ that satisfies φ under the valuation v;
- the set $V^*(\mathcal{A}, \varphi)$ of the parameter valuations $v \in \mathcal{D}(\varphi)$ such that all the timed sequences in $L(\mathcal{A}, v)$ satisfy φ under the valuation v.

For each of above sets, we study the *emptiness*, that is the problem of checking whether, given a PMITL formula φ (resp., given a PMITL formula φ and an L/U automaton $\mathcal{A}$) the set $X(\varphi)$, for $X \in \{S, V\}$, (resp., $X(\mathcal{A}, \varphi)$, for $X \in \{S^*, V^*\}$) contains any parameter valuation at all. Analogously, we consider the *universality*, that is the problem of checking whether given a PMITL formula (respectively, given a PMITL formula and an L/U automaton) the set contains all the admissible parameter valuations for the input formula.

We denote with the term *S-set* either $S(\varphi)$ or $S^*(\mathcal{A}, \varphi)$, for some PMITL formula φ and some L/U automaton $\mathcal{A}$. Analogously, we denote with *V-set* either $V(\varphi)$ or $V^*(\mathcal{A}, \varphi)$.

Restrictions on parameters. Observe that we have defined PMITL by imposing some restrictions on the parameters. First, we require the sets of parameters L and U to be disjoint. Second, we force each interval to have at most one parameter, either in the left or in the right endpoint. Third, in the definition of the decision problems we use admissibility for parameter valuations such that a parameterized interval cannot be evaluated neither as an empty nor a singular set. In Section 8, we discuss the impact of the first two restrictions on the decidability of the considered problems. In particular, we show that relaxing any of the first two restrictions leads to undecidability. Concerning the notion of admissibility of parameter valuations, the restriction to non-empty intervals seems reasonable since evaluating an interval to an empty set would cancel a portion of our specification (which would remain unchecked). Finally, the restriction to non-singular sets, which equals to disallow equality in the constraints, is already present in the temporal logic MITL [3], the parameterized discrete-time logic PLTL [2] and the parametric timed automata [17,7], and in all these formalisms it is crucial for achieving decidability. The same arguments used there can be applied here to show undecidability of the decision problems for PMITL if this restriction is relaxed.

4. Preliminary results

In this section we prove some results on the duality and the polarity of the parameterized operators which give further insights on our formalism and are used later in the paper. We also present a transformation of PMITL formulas that shifts the admissible domain for each parameter not occurring in constraints of the form $(c + z, d)$, to match the whole set $\mathbb{N}$.

Polarity of parameterized temporal operators. A temporal operator of PMITL is *upward-closed* if the interval in its subscript has a parameter from U in one of its endpoints. Analogously, a temporal operator of PMITL is *downward-closed* if the interval in its subscript has a parameter from L in one of its endpoints. The meaning of these definitions is clarified by the following lemma.

Let $\approx \in \{\leq, \geq\}$. Given two parameter valuations v and v', and a parameter z, with $\approx_z$ we denote the relation defined as: $v \approx_z v'$ iff $v(z) \approx v'(z)$ and $v(z') = v'(z')$ for any other parameter $z' \neq z$.

Lemma 1. *Let φ be the PMITL formula, $z \in U \cup L$, α be a timed sequence, and v and v' be parameter valuations.*

1. *For $z \in U$ and $v \leq_z v'$, if $(\alpha, v) \models \varphi$ then $(\alpha, v') \models \varphi$.*
2. *For $z \in L$ and $v \geq_z v'$, if $(\alpha, v) \models \varphi$ then $(\alpha, v') \models \varphi$.*

Proof. Consider first part (1) and suppose that $z \in U$. We show that for all $t \geq 0$, it holds that $(\alpha, v, t) \models \varphi$ implies $(\alpha, v', t) \models \varphi$. The proof is by structural induction on φ. The induction basis consists of the trivial case when z does not occur in φ. For the induction step, we only show in detail the cases $\varphi = \psi' \mathcal{U}_I \psi''$ and $\varphi = \psi' \mathcal{R}_I \psi''$, where I is a parameterized interval over the parameter z. All the other cases are simpler.

In the first case, since $z \in U$, I is of the form $(c, d + z)$. From $(\alpha, v, t) \models \varphi$, we get that I_v is not empty and there is $t' \in I_v + t$ such that $(\alpha, v, t') \models \psi''$ holds, and $(\alpha, v, t'') \models \psi'$ holds for all $t < t'' < t'$. By induction hypothesis, we get that $(\alpha, v', t') \models \psi''$ holds, and $(\alpha, v', t'') \models \psi'$ also holds for all $t < t'' < t'$. Since $v(z) \leq v'(z)$, $I_v \subseteq I_{v'}$, and thus we clearly get that $(\alpha, v', t) \models \varphi$.

In the second case, I is of the form $(c+z, d)$. From $(\alpha, v, t) \models \varphi$, we get that either for all $t' \in I_v + t$, $(\alpha, v, t') \models \psi''$ holds, or there is a $t'' > t$ such that $(\alpha, v, t'') \models \psi'$ holds and $(\alpha, v, t') \models \psi''$ holds for all $t' \in [t, t''] \cap I_v + t$. Note that now $I_{v'} \subseteq I_v$. Therefore, if $(\alpha, v, t') \models \psi''$ holds for all $t' \in I_v + t$, then by induction hypothesis, also $(\alpha, v', t') \models \psi''$ holds for all $t' \in I_{v'} + t$. In the other case, i.e., there is a $t'' > t$ such that $(\alpha, v, t'') \models \psi'$ holds and $(\alpha, v, t') \models \psi''$ holds for all $t' \in [t, t''] \cap I_v + t$, again by induction hypothesis, also $(\alpha, v', t'') \models \psi'$ holds and $(\alpha, v', t') \models \psi''$ holds for all $t' \in [t, t''] \cap I_{v'} + t$. Thus $(\alpha, v', t) \models \varphi$.

Part (2) can be shown using similar arguments, and thus we omit the details. □

Negation and duality. Formulas of PMITL are in *negation normal form*, that is negation occurs only at the level of the atomic propositions. This is standard for parameterized linear temporal logic, since the negation inverts the polarity of the parameterized temporal operators occurring within the scope of the negation operator. Nevertheless, our logic is closed under semantic negation.

For a PMITL formula φ, denote with $\sim\varphi$ the PMITL formula obtained from φ by replacing each operator with its dual (i.e., exchanging the $\vee$'s with the $\wedge$'s and the $\mathcal{U}$'s with $\mathcal{R}$'s) and negating the atomic propositions.

Directly derived from the semantics, the duality of the temporal operators of PMITL follows, that is it holds that $\neg(\varphi \mathcal{U}_I \psi) \equiv \neg\varphi \mathcal{R}_I \neg\psi$, for any interval I. Thus, using De Morgan's laws, the following proposition holds.

Proposition 2. *For any PMITL formula φ:* $\quad \neg\varphi \equiv \sim\varphi$.

Observe that if I is a parameterized interval, then the operators $\mathcal{U}_I$ and $\mathcal{R}_I$ have opposite polarities, and then in $\sim\varphi$ the role of the sets L and U is exchanged (i.e., $\sim\varphi$ is a PMITL formula over the set of lower bound parameters U and the upper bound parameters L), and in particular the following holds.

Proposition 3. *For any PMITL formula φ, φ is a* PMITL$_\Diamond$ *formula if and only if $\sim\varphi$ is a* PMITL$_\Box$ *formula.*

Finally, the satisfiability of $\sim\varphi$ is dual with respect to the validity of φ. In fact, one can observe that φ and $\sim\varphi$ have the same set of admissible parameter valuations and $V(\varphi)$ is the complement of $S(\sim\varphi)$, with respect to set of the admissible parameter valuations, and analogously, $V^*(\mathcal{A}, \varphi)$ is the complement of $S^*(\mathcal{A}, \sim\varphi)$. Thus, we get the following lemma.

Lemma 2. *The emptiness/universality problems of V-sets for a PMITL formula φ reduce in linear time to the universality/emptiness problems of S-sets for $\sim\varphi$, and vice-versa.*

Proof. As observed, $\mathcal{D}(\sim\varphi) = \mathcal{D}(\varphi)$, since the subscripted intervals in φ and $\sim\varphi$ are the same and, moreover, the formula $\sim\varphi$ has the same size as φ. Since $V(\varphi) = \mathcal{D}(\varphi) \setminus S(\sim\varphi)$, then $V(\varphi) = \emptyset$ if and only if $S(\sim\varphi) = \mathcal{D}(\sim\varphi)$, and $V(\varphi) = \mathcal{D}(\varphi)$ if and only if $S(\sim\varphi) = \emptyset$.

Analogously, $V^*(\mathcal{A}, \varphi) = \mathcal{D}(\varphi)$ if and only if $S^*(\mathcal{A}, \sim\varphi) = \emptyset$, and $V^*(\mathcal{A}, \varphi) = \emptyset$ if and only if $S^*(\mathcal{A}, \sim\varphi) = \mathcal{D}(\sim\varphi)$.

The formula $\sim\varphi$ has been obtained by simply dualizing the operators in φ, and thus the statement follows. □

Normalization of intervals.

A PMITL formula φ is *well defined* if $c < d$ for all its parameterized intervals of the form $(c, d + z)$. Observe that, if the formula is well defined, and z is one of its parameters not occurring in any constraint of the form $(c + z, d)$, then each non-negative number can be assigned to z by an admissible valuation. The following lemma shows that it suffices to solve the introduced decision problems for well defined PMITL formulas.

For parameter valuations v, v', we denote with $v - v'$ the function that maps each parameter z to $v(z) - v'(z)$.

Lemma 3. *Given a PMITL formula φ, there exist a well defined PMITL formula φ' and a parameter valuation v_0 such that $N_{\varphi'} = N_\varphi$, $K_{\varphi'} = O(K_\varphi)$, and, for each parameter valuation v, called $v' = v - v_0$:*

- *$v \in \mathcal{D}(\varphi)$ if and only if $v' \in \mathcal{D}(\varphi')$,*
- *for each timed sequence α and for each parameter valuation v: $(\alpha, v) \models \varphi$ if and only if $(\alpha, v') \models \varphi'$.*

Proof. Fix a PMITL formula φ. For each parameter z appearing in φ, define the constant m_z as:

$$m_z = \begin{cases} \max\{c - d + 1 | (c, d + z) \text{ is in } \varphi\} & \text{if } \max\{c - d + 1 | (c, d + z) \text{ is in } \varphi\} > 0 \\ 0 & \text{otherwise.} \end{cases}$$

Observe that $m_z \leq v(z)$, for any parameter valuation $v \in \mathcal{D}(\varphi)$.

Let φ' be the PMITL formula obtained from φ by replacing each occurrence of a parameter z with $m_z + z$. Thus, each interval of the form $(c, d + z)$ is replaced with $(c, d + m_z + z)$, and each interval of the form $(a + z, b)$ with $(a + m_z + z, b)$. Clearly, from the definition of the constants m_z, we get that φ' is well defined.

Denote with v_0 the parameter valuation that maps each parameter z to m_z and let $v \in \mathcal{D}(\varphi)$. Since $v(z) \geq m_z$, for each parameter z, $v - v_0$ is a parameter valuation (i.e., it assigns a natural number to each parameter). Moreover, observe that an admissibility constraint $v(z) > c - d$, which derives from an interval $I = (c, d + z)$ in φ, is replaced with $v'(z) > c - d - m_z$, from the interval $I' = (c, d + m_z + z)$ in φ'. Analogously, an admissibility constraint $v(z) < b - a$, deriving from $I = (a + z, b)$ in φ, is replaced with $v'(z) < b - a - m_z$ from $I' = (a + m_z + z, b)$ in φ'. Thus, for $v' = v - v_0$, we have that $v' \in \mathcal{D}(\varphi')$ if and only if $v \in \mathcal{D}(\varphi)$. Moreover, for any choice of I and I' as above, $I_v = I'_{v'}$ holds. Thus, we have that α satisfies φ under valuation v if and only if α satisfies φ' under valuation v'.

Now, let M be the maximum value of the defined m_z. Clearly, there is a constant c in the formula φ such that M is less than or equal to $c + 1$, and then $M \le K_\varphi$ (recall that K_φ is the maximal constant used in φ augmented by 1). Moreover, the constants in φ' exceed those in φ of at most M, therefore $K'_\varphi \le 2K_\varphi$ holds.

Finally, φ' has the same number of the subformulas as φ, and this completes the proof. $\square$

Directly from the above lemma, the following reduction holds.

Lemma 4. *The emptiness/universality problems of S-sets (resp. V-sets) for PMITL formulas reduce in linear time to the universality/emptiness problems of S-sets (resp. V-sets) for well defined PMITL formulas.*

5. Decidability of PMITL

In this section, we prove that the satisfiability, validity, and model-checking problems defined in Section 3 are decidable and Expspace-complete thus matching the computational complexity for MITL formulas [3]. A central step in our argument is a translation to the emptiness and the universality problems for Büchi L/U automata.

First we show a normal form for well defined PMITL formulas. A PMITL formula is in *normal form* if each of its temporal subformulas is of one of the following types:

1. $\Diamond_{]0,b[}\varphi'$, or $\Diamond_{]0,b]}\varphi'$, where $b \in \mathbb{N}$;
2. $\Box_{]0,b[}\varphi'$, or $\Box_{]0,b]}\varphi'$, where $b \in \mathbb{N}$;
3. $\varphi_1\mathcal{U}_{(a,b)}\varphi_2$, or $\varphi_1\mathcal{R}_{(a,b)}\varphi_2$, where $a, b \in \mathbb{N}$ and $a > 0$;
4. $\varphi_1\mathcal{U}_{]0,\infty[}\varphi_2$;
5. $\Box_{]0,\infty[}\varphi'$;
6. $\Diamond_{(0,x)}\varphi'$, where $x \in U$;
7. $\Box_{(0,y)}\varphi'$, where $y \in L$;
8. $\Box_{(a+x,b)}\varphi'$, where $a, b \in \mathbb{N}$, $x \in U$;
9. $\Diamond_{(a+y,b)}\varphi'$, where $a, b \in \mathbb{N}$, $y \in L$.

We start by showing the translations which allow to obtain the normal form for a given PMITL formula.

Lemma 5. *For every well defined PMITL formula φ, there is an equivalent PMITL formula ψ using only $\Diamond$ and $\Box$ as parameterized temporal operators, and using only parameterized intervals of the form $(0, z)$ or $(c + z, d)$, for $c, d \in \mathbb{N}$ and $z \in U \cup L$. Moreover, $N_\psi = O(N_\varphi)$ and $K_\psi = K_\varphi$.*

Proof. The first step consists in showing that the parameterized operators $\Diamond_H$ and $\Box_J$ are sufficient to define all the parameterized operators of PMITL. This can be achieved by rewriting a well defined PMITL formula according to the following equivalences:

1. $\varphi\mathcal{U}_{(c,d+z)}\psi \equiv (\varphi\mathcal{U}_I\psi) \wedge \Diamond_{(c,d+z)}\psi$,
2. $\varphi\mathcal{R}_{(c,d+z)}\psi \equiv (\varphi\mathcal{R}_I\psi) \vee \Box_{(c,d+z)}\psi$,
3. $\varphi\mathcal{U}_{(c+z,d)}\psi \equiv \Box_{[0,z]}(\varphi\mathcal{U}_I\psi) \wedge \Diamond_{(c+z,d)}\psi$,
4. $\varphi\mathcal{R}_{(c+z,d)}\psi \equiv \Diamond_{[0,z]}(\varphi\mathcal{R}_I\psi) \vee \Box_{(c+z,d)}\psi$,

where $I = (c, \infty[$ and: for the equivalences 1 and 2, I is left-closed if and only if $(c, d + z)$ is left-closed, and for the equivalences 3 and 4, I is left-closed if and only if $(c + z, d)$ is left-closed.

Thanks to the duality of temporal operators, the second equivalence can be obtained from the first one and the fourth one from the third one.

To see the equivalence 1, observe that by definition, $\varphi\mathcal{U}_{(c,d+z)}\psi$ is satisfied on a timed sequence α under a valuation v if and only if there is a $t \in (c, d + v(z))$ such that $(\alpha, v, t) \models \psi$ and for all positive $t' < t$, $(\alpha, v, t') \models \varphi$. This is equivalent to require that both: (1) there is a $t \ge c$ (or $t > c$, if I is left-open) such that $(\alpha, v, t) \models \psi$ and for all positive $t' < t$, $(\alpha, v, t') \models \varphi$, and (2) there is a $t \in (c, d + v(z))$ such that $(\alpha, v, t) \models \psi$, and thus the equivalence holds.

For the equivalence 3, suppose that $(\alpha, v) \models \varphi\mathcal{U}_{(c+z,d)}\psi$. From the semantics, clearly $(\alpha, v) \models \Diamond_{(c+z,d)}\psi$ also holds. Moreover, there must be a $t \in (c + v(z), d)$ such that $(\alpha, v, t) \models \psi$ and for all positive $t' < t$, $(\alpha, v, t') \models \varphi$. For all $t'' \in [0, v(z)]$ we get that $t - t'' \in I$, and thus also $(\alpha, v) \models \Box_{[0,z]}(\varphi\mathcal{U}_I\psi)$ holds. For the converse direction, from $(\alpha, v) \models \Box_{[0,z]}(\varphi\mathcal{U}_I\psi)$, we get that there is a $t \in I$ such that $(\alpha, \nu, \nu(z)+t) \models \psi$ and $(\alpha, \nu, t') \models \varphi$ for each positive $t' < t$. Since $\nu(z)+t \in (c+\nu(z), \infty[$ and $(\alpha, \nu) \models \Diamond_{(c+z,d)}\psi$ also holds, we can assume that $\nu(z)+t \in (c+\nu(z), d)$ and thus, $(\alpha, \nu) \models \varphi\mathcal{U}_{(c+z,d)}\psi$.

Note that the parameters in all the above equations are used with the same polarity and without changing the set of admissible values.

The second step consists of removing all the intervals of the $(c, d+z)$ from the operators $\Box$ and $\Diamond$. Note that by hypothesis the formula is well defined, and then $c \leq d$ and we can safely use intervals of the form (c, d) and $[0, z]$ to do so. The following equivalences can be directly derived from the semantics:

1. $\Diamond_{(c,d+z)}\varphi \equiv \Diamond_{(c,d)}\Diamond_{[0,z]}\varphi$,
2. $\Box_{(c,d+z)}\varphi \equiv \Box_{(c,d)}\Box_{[0,z]}\varphi$,

where (c, d) is left/right open if $(c, d+z)$ is left/right open. Again, the polarity of the parameters in the above equivalences is the same on both sides of the same equivalence.

Observe that the transformations described by the above equivalences keep unchanged the maximal constant of the formula and introduce at most a constant number of new subformulas for each subformula of the starting formula. Therefore, we can rewrite a PMITL formula φ into an equivalent formula ψ such that $K_\psi = K_\varphi$ and $N_\psi = O(N_\varphi)$ and thus the lemma holds. $\Box$

For the above lemma, we can consider PMITL formulas satisfying normal form conditions with respect to all the parameterized operators. Observe that the normal form defined in [3] coincides with our normal form for the subformulas not containing parameterized operators. Moreover, the transformations used there to reduce in normal form each MITL formula can be applied to PMITL formulas without altering the polarity of the parameterized temporal operators. Therefore we can apply them to the PMITL formulas as resulting from Lemma 5, thus obtaining an equivalent PMITL formula in normal form.

Lemma 6. *Given a well defined PMITL formula φ, there is an equivalent PMITL formula ψ in normal form such that $N_\psi = O(N_\varphi)$ and $K_\psi = K_\varphi$. Moreover the translation is performed in linear time.*

When a PMITL formula does not contain parametric intervals of the form $(c+z, d)$, we are able to adapt the constructions given in [3] for MITL and in [7] for $\mathrm{P}_{0,\infty}\mathrm{MITL}_{0,\infty}$ to obtain an equivalent PTA. This result is stated in the following theorem.

Theorem 2. *Given a PMITL formula φ in normal form, which does not contain intervals of the form $(c+z, d)$, one can construct a Büchi L/U automaton $\mathcal{A}_\varphi$ accepting a timed sequence α under a parameter valuation ν if and only if $(\alpha, \nu) \models \varphi$. Also, $\mathcal{A}_\varphi$ has $O(2^{N_\varphi \times K_\varphi})$ locations, $O(N_\varphi \times K_\varphi)$ clocks, and constants bounded by K_φ. Moreover, if φ has only parameters from the set L (resp. U) the resulting automaton $\mathcal{A}_\varphi$ also has only lower bound parameters from L (resp. upper bound parameters from U).*

Proof. Fix a PMITL formula φ. By [3], for each parameter valuation ν and timed sequence α, there is a timed sequence $\alpha_{\varphi,\nu}$ which is equivalent to α such that for each subformula ψ of φ, for each interval I of $\alpha_{\varphi,\nu}$ and $t, t' \in I$: $(\alpha, \nu, t) \models \psi$ iff $(\alpha, \nu, t') \models \psi$. Thus, for a given parameter valuation ν, we can restrict ourselves to consider only such timed sequences.

Observe that the construction given in [3] defines the behaviors of the automaton modularly to each subformula of the input formula. Therefore, we only need to augment that construction with the portion concerning the subformulas corresponding to the parameterized operators.

Since φ is in normal form, the only subformulas of φ with parameterized operators are of the form $\Diamond_{(0,x)}\chi$ and $\Box_{(0,y)}\chi$.

We fix a parameter valuation ν and consider first a subformula ψ of φ of the form $\Diamond_I\chi$, for $I = (0, x)$. To check the fulfillment of ψ, we need to use one clock which we call ξ, along with the clock constraint $\xi \in I$. As in [3], 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 ν). In order to witness the fulfillment of ψ at the current time t, the automaton resets ξ and stores in its finite control the obligation that χ must hold at a time $t+d$, such that $d \in I_\nu$. The obligation is discharged as soon as an appropriate state is found, keeping track that χ holds at the current time: in this case we have that ψ is fulfilled. We cannot reset ξ if an obligation for ψ is already pending and cannot be discharged, 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 ξ 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 $\xi \in I$ 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 $d \geq \nu(x)$ (or $d > \nu(x)$). The behavior we have described needs to be implemented to check the truth of ψ over intervals and there are some subtleties concerning the treatment of open intervals that have been omitted here because the details on these aspects do not differ from the case of formulas of the form $\Diamond_{(0,b)}\chi$, where b is a constant, which is carefully explained in [3].

A subformula ψ of φ of the form $\Box_{(0,y)}\chi$ is dual to formula of the form $\Diamond_{(0,x)}\chi$. Therefore, we can argue similarly that an automaton can check for the fulfillment of ψ, simply using a clock ξ and a clock constraint $\xi > y$, if the interval is right-closed, and $\xi \geq y$ if the interval is right-open.

From [3], we get that the non-parameterized portion of the construction of $\mathcal{A}_\varphi$ has $O(2^{N_\varphi \times K_\varphi})$ locations, $O(N_\varphi \times K_\varphi)$ clocks, and constants bounded by K_φ. Clearly, the addition we have described above does not alter these measures.

To complete the proof, observe that, the constructed $\mathcal{A}_\varphi$ uses exactly the parameters of the formula φ such that 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 $\mathcal{A}_\varphi$. $\square$

We can now show the main result of this section.

Theorem 3. *The problems of checking the emptiness and the universality of each of S-sets are* Expspace*-complete.*

Proof. Hardness follows from Expspace-hardness of both satisfiability and model-checking problems for MITL formulas [3].

For the membership, by Lemma 4, we can consider a well defined formula. Moreover, by Lemma 6, we can transform, in linear time, a given well defined formula into an equivalent one in normal form. Then we fix a PMITL formula φ in normal form.

Let us start with the emptiness problem for $S(\varphi)$. Observe that if φ contains a parameterized interval of the form $(c+z, d)$ where $c \geq d$ then $\mathcal{D}(\varphi)$ is empty and thus $S(\varphi) = \emptyset$. In the remaining case, we use the following algorithm. First, assign each parameter $x \in L$ appearing in a subformula of φ of the form $\Diamond_{(c+x,d)}\psi$, $c < d$, with the minimum value assigned by an admissible parameter valuation, and each parameter $y \in U$ in a subformula of φ of the form $\Box_{(c'+y,d')}\psi'$, $c' < d'$, with the maximum value assigned by an admissible parameter valuation. Note that these values are well defined since for such parameters the admissible values are bounded (in fact, from the first kind of formula we get the constraint $0 \leq x \leq d-c-1$, and for the second the constraint $0 \leq y \leq d'-c'-1$).

Now, denote with φ' the resulting formula, construct the Büchi L/U automaton $\mathcal{A}_{\varphi'}$ as in Theorem 2 and then check $\Gamma(\mathcal{A}_{\varphi'})$ for emptiness. Since to obtain φ' we assign with the minimum admissible value only parameters of L and with the maximum admissible value only parameters of U, by Lemma 1 we get that $S(\varphi)$ is empty if and only if $S(\varphi')$ is empty. Moreover, by Theorem 2, $S(\varphi')$ is empty if and only if $\Gamma(\mathcal{A}_{\varphi'})$ is empty, and thus the above algorithm correctly checks $S(\varphi)$ for emptiness. By Theorem 2, $\mathcal{A}_{\varphi'}$ has exponential size in $K_{\varphi'}$ and $N_{\varphi'}$. From the construction above, the number of subformulas of φ' and the constants occurring as subscripts in φ' are at most those of φ. Moreover, by Theorem 1, checking the emptiness of $\Gamma(\mathcal{A}_{\varphi'})$ requires polynomial space in the size of $\mathcal{A}_{\varphi'}$, and thus we get that the above algorithm takes exponential space.

For the universality problem for $S(\varphi)$ we can reason analogously. First, assign each parameter x appearing in a subformula of φ of the form $\Diamond_{(c+x,d)}\psi$ with the maximum value assigned by an admissible parameter valuation, and each parameter y in a subformula of φ of the form $\Box_{(c'+y,d')}\psi'$ with the minimum value assigned by an admissible parameter valuation. Now, denote with φ' the resulting formula, construct the Büchi L/U automaton $\mathcal{A}_{\varphi'}$ as in Theorem 2 and then check $\Gamma(\mathcal{A}_{\varphi'})$ for universality. Thus, by Theorem 1, we get that also this problem is in Expspace.

Finally, membership in Expspace of deciding emptiness and universality of $S^*(\varphi, \mathcal{A})$ can be shown with similar arguments, we just need to change the last step of the above algorithm to check the desired property of the set Γ for the intersection of the given $\mathcal{A}$ and the constructed $\mathcal{A}_{\varphi'}$, and not just for $\mathcal{A}_{\varphi'}$.

By Theorem 3 and Lemma 2 the following result also holds.

Corollary 1. *The problems of checking the emptiness and the universality of V-sets are* Expspace*-complete.*

6. Computational complexity in fragments of PMITL

In this section, we address the complexity of the parameterized operators in PMITL, and thus of the corresponding logic fragments. We focus on fragments of $\text{PMITL}_{0,\infty}$ which include $\text{MITL}_{0,\infty}$ (recall that satisfiability and model-checking problems for $\text{MITL}_{0,\infty}$ are known to be Pspace-hard [3]). On the positive side, we prove that the emptiness problems for $\text{PMITL}_\Diamond$ formulas and the universality problems for $\text{PMITL}_\Box$ formulas are in Pspace. We also show Expspace-hardness of the remaining problems in $\text{PMITL}_\Diamond$ and $\text{PMITL}_\Box$, and of all the considered problems in the fragments of PMITL allowing subformulas either of the form $\Box_{(c+x,d)}$ or of the form $\Diamond_{(c+y,d)}$. We start giving such hardness results.

Lemma 7. *The following problems are* Expspace*-hard:*

1. *Deciding the universality of the S-sets and V-sets for* $\text{PMITL}_\Diamond$ *formulas.*
2. *Deciding the emptiness of S-sets and V-sets for* $\text{PMITL}_\Box$ *formulas.*
3. *Both deciding emptiness and deciding the universality of S-sets and V-sets for formulas whose parameterized operators are either of the form $\Box_{(c+y,d)}$ or of the form $\Diamond_{(c+x,d)}$.*

Proof. We only give the proofs for the set $S(\varphi)$. The proofs for $S^*(\mathcal{A}, \varphi)$ can be obtained reducing the corresponding results for $S(\varphi)$ by considering the automaton $\mathcal{A}$ accepting all timed sequences. Concerning the V-sets, we can reduce the dual problems for the S-sets by Lemma 2 and Proposition 3. Consider of fragments of MITL where intervals of the form $(c, c+1)$ are allowed either only on the operator $\Diamond$ or only on the operator $\Box$, and the rest of the intervals (a, b), used in the subscripts of the temporal operators, are such that either $a = 0$ or $b = \infty$. The satisfiability problem in each such fragment is known to be Expspace-complete (see [3]).

Consider any MITL formula φ from the above fragment where $(c, c+1)$ is allowed as subscript only of the operator $\Diamond$. Rewrite φ to a formula φ' with a parameter $x \in U$, where each operator of the form $\Diamond_{(c,c+1)}$ is replaced with $\Diamond_{(c,c+1+x)}$ and any other part of the formula stays unchanged. We claim that φ is satisfiable if and only if $S(\varphi')$ is universal. First observe that all the parameter valuations are admissible, and therefore, "$S(\varphi')$ is universal" means that $S(\varphi') = \mathbb{N}$. Thus, if $S(\varphi')$ is universal then $0 \in S(\varphi')$, and hence φ is satisfiable (0 denotes the valuation assigning 0 the x). Vice-versa, if φ is satisfiable then $0 \in S(\varphi')$, and by Lemma 1 we get that $S(\varphi')$ is universal. Therefore, checking the universality of $S(\varphi)$ is Expspace-hard.

To show Expspace-hardness of the emptiness problem of S-sets for $\mathrm{PMITL}_\Box$ formulas we consider a MITL formula φ from the above fragment where $(c, c+1)$ is allowed as subscript only of the operator $\Box$. Rewrite φ to a formula φ' where each operator of the form $\Box_{(c,c+1)}$ is replaced with $\Box_{(c,c+1+y)}$, for a parameter $y \in L$, and any other part of the formula stays unchanged. Thus, if φ is satisfiable then trivially $S(\varphi')$ is not empty (it contains at least 0). Vice-versa, if $S(\varphi')$ is not empty, then 0 must belong to $S(\varphi')$, by Lemma 1, and therefore φ is satisfiable.

For the third item, again we reduce the satisfiability problem for the MITL fragments considered above. In particular, we rewrite each operator of the form $\Diamond_{(c,c+1)}$ with $\Diamond_{(c+y,c+1)}$, for a parameter $y \in L$, and any other part of the formula stays unchanged. In this case, we observe that the only admissible value for y is 0, thus testing the emptiness of $S(\varphi')$ coincides with testing its universality and φ is satisfiable if and only if $S(\varphi')$ is not empty. Therefore, the claimed result again follows from the Expspace-hardness of satisfiability for the considered fragment of MITL. The case of the operator $\Box_{(c+x,d)}$, for $x \in U$, is analogous, and thus we omit further details. $\Box$

From the above Lemma 7 and Theorem 3, we can state the following theorem.

Theorem 4. *The problems of checking the emptiness and the universality of each of S-sets and V-sets, restricted to* $\mathrm{PMITL}_{0,\infty}$ *formulas, are* Expspace*-complete.*

The hardness results from Lemma 7 leave open the computational complexity for some of the considered decision problems in the fragments $\mathrm{PMITL}_\Diamond$ and $\mathrm{PMITL}_\Box$. In the rest of this section, we show that such decision problems are indeed Pspace-complete. This is an interesting result, since these fragments include $\mathrm{P}_{0,\infty}\mathrm{MITL}_{0,\infty}$ and capture meaningful properties (see the example from Section 3).

For a sequence α, we denote with $S_\alpha(\varphi)$ the set of parameter valuations v such that the sequence α satisfies φ under valuation v. Observe that $S(\varphi) = \bigcup_\alpha S_\alpha(\varphi)$ holds.

We start by showing Pspace membership for the emptiness problems in $\mathrm{PMITL}_\Diamond$. The next lemma is the crucial result for reducing such problems for the S-sets to the same problems in $\mathrm{P}_{0,\infty}\mathrm{MITL}_{0,\infty}$.

Lemma 8. *Let φ be a well defined* $\mathrm{PMITL}_\Diamond$ *formula containing a subformula χ of the form $\Diamond_I \psi$, for $I = (c, d+z)$. Denote with φ' the formula obtained from φ by replacing χ with the subformula $\chi' = \Box_{]0,c]} \Diamond_{I-c} \psi$. For each timed sequence α, the following properties hold:*

1. $S_\alpha(\varphi') \subseteq S_\alpha(\varphi)$.
2. $S_\alpha(\varphi) = \emptyset \Longleftrightarrow S_\alpha(\varphi') = \emptyset$.

Proof. Consider first part (a). Assume that $S_\alpha(\varphi') \neq \emptyset$, otherwise the assertion is trivially true. The proof proceeds by structural induction on the possible forms of φ.

Consider the basis case $\varphi = \chi$, i.e. φ has the form $\Diamond_I \psi$, for $I = (c, d+z)$. Let us consider only the case that I is the closed interval $[c, d+z]$, the other cases being analogous. If $v \in S_\alpha(\varphi')$, for all $t \in]0, c]$, the subformula ψ must be true for some $t' \in [t, t+d+v(z)-c]$, and in particular, ψ must be true in the interval $[c, d+v(z)]$. This suffices to conclude that $v \in S_\alpha(\varphi)$, too.

The induction step involving boolean operators is quite simple. If $\varphi = \varphi_1 \wedge \varphi_2$, and χ is a subformula of φ_1, clearly we get that $\varphi' = \varphi_1' \wedge \varphi_2$, where φ_1' denotes the formula obtained from φ_1 by replacing χ with χ', and by induction hypothesis $S_\alpha(\varphi_1') \subseteq S_\alpha(\varphi_1)$. Therefore, $S_\alpha(\varphi') = S_\alpha(\varphi_1') \cap S_\alpha(\varphi_2) \subseteq S_\alpha(\varphi_1) \cap S_\alpha(\varphi_2) = S_\alpha(\varphi)$. The disjunction of two subformulas is analogous.

If the main operator of φ is a temporal operator, as a sample case, consider φ of the form $\varphi_1 \mathcal{U}_I \varphi_2$ and χ is a subformula of φ_1. Let φ_1' denote the formula obtained from φ_1 by replacing χ with χ'. From $v \in S_\alpha(\varphi')$, we get that $(\alpha, v) \models \varphi'$. From the semantics this means that there is a time $t \in I_v$ such that $(\alpha, v, t) \models \varphi_2$ and $(\alpha, v, t') \models \varphi_1'$ for all $0 < t' < t$. By induction hypothesis, $S_\beta(\varphi_1') \subseteq S_\beta(\varphi_1)$ for each β, thus also for each timed sequence $\alpha_{t'}$ that matches the suffix of α from t'. Therefore, $(\alpha, v, t') \models \varphi_1$ for all $0 < t' < t$, and thus $(\alpha, v) \models \varphi$.

The remaining cases are treated analogously.

To prove part (b) of (1), observe that directly from part (a) of (1), we get that $S_\alpha(\varphi) = \emptyset$ implies $S_\alpha(\varphi') = \emptyset$. For the other direction, we show a stronger result. If $v \in S_\alpha(\varphi)$ then $v' \in S_\alpha(\varphi')$ where v' is defined as $v'(z) = v(z) + c$ and $v'(y) = v(y)$ for $y \neq z$. We can prove this result by structural induction on the formulas φ which contain χ as a subformula.

The base case is $\varphi = \chi$. From $v \in S_\alpha(\varphi)$, we get that the subformula ψ is true along α starting at some time $t' \in [c, d + v(z)]$. Clearly, for all $0 < t \le c$, we have that $t \le t'$ and $t' \le d + v(z)$. Thus, $t \le t' \le d + v'(z) - c$. From $t > 0$, we get that $t \le t' \le d + v'(z) - c + t$. Recall that $I - c = [0, d - c + z]$. Therefore, $\diamond_{[0,d-c+z]}\, \psi$ holds at all $t \in]0, c]$. Thus $(\alpha, v') \models \varphi'$.

For the induction step, the cases involving Boolean operators are quite straightforward from the semantics of the operator as in the proof of part (a) and thus we omit further details on them. For the cases involving the temporal operators the key observation is to exploit the polarity of the temporal operators using z (which is in U). Therefore, for each subformula ψ, by Lemma 1, we get that $v' \in S_\beta(\psi)$ whenever $v \in S_\beta(\psi)$, for each timed sequence β. The rest of the proof simply uses the semantics of the operators and the induction hypothesis, thus we omit further details. □

Fix a formula φ of PMITL$_\diamond$. By applying transformations as in the above Lemma 8 to φ, from inside out, we can generate a sequence of formulas $\varphi = \varphi_1, \ldots, \varphi_n = \varphi'$ where φ' is a $\mathrm{P}_{0,\infty}\mathrm{MITL}_{0,\infty}$ formula and n is at most the number of sub-formulas of φ. Since at each step of the transformation we can apply the above lemma, we get that the properties (a) and (b) stated in the Lemma 8 also holds for such φ and φ'. Moreover, observe that each transformation only adds one subformula, therefore the size φ' is linear in the size of φ.

From the property (b), it is correct to check the emptiness of $S(\varphi)$ by checking the emptiness of $S(\varphi')$, and thus the described transformations give a reduction of such problem for PMITL$_\diamond$ formulas to the same problem in $\mathrm{P}_{0,\infty}\mathrm{MITL}_{0,\infty}$. Since the latter one is known to be in PSPACE [7], we get PSPACE membership also for the problem of checking the emptiness of the $S(\varphi)$ sets in PMITL$_\diamond$. We can repeat the same arguments for showing PSPACE-membership of checking the emptiness of the $S(\mathcal{A}, \varphi)$ sets for a given PTA $\mathcal{A}$. Moreover, by Proposition 3 and Lemma 2, we get also PSPACE-membership of the universality problems for the V-sets in PMITL$_\Box$. Therefore the following theorem holds.

Theorem 5. *The following problems are* PSPACE*-complete:*

1. *Checking the emptiness of each of the S-sets, restricted to input formulas from* PMITL$_\diamond$.
2. *Checking the universality of each of the V-sets, restricted to input formulas from* PMITL$_\Box$.

Now, we turn our attention to the universality problems for the S-sets in PMITL$_\Box$. We omit the proof of the next lemma, since it is very similar to the proof of Lemma 8, and in fact what we prove here is essentially dual to what is shown there. Observe that part (b) of the following lemma is indeed dual to the assertion used to prove part (b) of Lemma 8.

Lemma 9. *Let φ be a well defined* PMITL$_\Box$ *formula containing a subformula χ of the form $\Box_I\, \psi$, for $I = (c, d + z)$. Denote with φ' the formula obtained from φ by replacing χ with the subformula $\chi' = \diamond_{]0,c]}\, \Box_{I-c}\, \psi$. For each timed sequence α, the following properties hold:*

a) $S_\alpha(\varphi) \subseteq S_\alpha(\varphi')$.
b) *For each constant $c' \ge c$ and for each parameter valuation $v' \in S_\alpha(\varphi')$ such that and $v'(z) > c'$, denoting with v the parameter valuation defined as $v(z) = v'(z) - c'$ and $v(y) = v'(y)$ for each $y \neq z$, $v \in S_\alpha(\varphi)$ holds.*

By replacing all the subformulas $\Box_I\, \psi$ in a PMITL$_\Box$ formula φ, as in the above Lemma 9, we can generate a sequence of formulas $\varphi = \varphi_1, \ldots, \varphi_n = \varphi'$ where the last one is a $\mathrm{P}_{0,\infty}\mathrm{MITL}_{0,\infty}$ formula. For such φ and φ' we prove the following lemma.

Lemma 10. *$S(\varphi)$ is universal if and only if $S(\varphi')$ is universal.*

Proof. Since the domain of the admissible valuations for φ and φ' is the same (i.e., $\mathcal{D}(\varphi) = \mathcal{D}(\varphi')$), from part (a) of Lemma 9 we trivially get the "only if" direction. For the converse direction, let $c_{\max}$ be the maximum over the constants c such that $\Box_{(c,d+z)}\, \psi$ is a subformula of φ, for some constant d and parameter z, and $L' = \{z_1, \ldots, z_n\}$ be the set of such z. Observe that, if $v' \in S_\alpha(\varphi')$, for a timed sequence α, is a parameter valuation such that $v'(z) > c_{\max}$ for all parameters $z \in L'$ and, for a given z, v is the parameter valuation defined as $v(z) = v'(z) - c_{\max}$ and $v(y) = v'(y)$ for each $y \neq z$, then, by part (b) of the lemma and Lemma 1, $v \in S_\alpha(\varphi)$ holds.

Let $v \in \mathcal{D}(\varphi)$ and define $v'(z) = v(z) + c_{\max}$ for all parameters $z \in L'$. Clearly, $v' \in \mathcal{D}(\varphi')$. Define a sequence of parameter valuations $v_0, \ldots, v_n$ such that $v_0 = v'$, $v_n = v$ and for $i = 1, \ldots, n$, v_i is defined such that $v_i(z_i) = v_{i-1}(z_i) - c_{\max}$ and $v_i(y) = v_{i-1}(y)$ for all $y \neq z_i$. By the above observation, we have that for a timed sequence α, if $v' \in S_\alpha(\varphi')$ then also $v \in S_\alpha(\varphi)$. Therefore, if $v' \in S(\varphi')$ then also $v \in S(\varphi)$, which proves the claim. □

Finally, observe that each transformation step from φ to φ' only adds one subformula, therefore φ' has size linear in the size of φ. From the above Lemma 10 the described transformation give a reduction of this problem for PMITL$_\Box$ formulas

Table 1
Summary of the computational complexity of the emptiness and universality problems for the sets $S(\varphi)$, $S^*(\mathcal{A},\varphi)$, $V(\varphi)$ and $V^*(\mathcal{A},\varphi)$ in the studied syntactic fragments of PMITL.

Logic	Emptiness	Universality
PMITL	EXPSPACE-complete	EXPSPACE-complete
$\text{PMITL}_{0,\infty}$	EXPSPACE-complete	EXPSPACE-complete
$\text{PMITL}_{\diamond}$	PSPACE-complete	EXPSPACE-complete
$\text{PMITL}_{\Box}$	EXPSPACE-complete	PSPACE-complete
$\text{P}_{0,\infty}\text{MITL}_{0,\infty}$	PSPACE-complete	PSPACE-complete

to the same problem in $\text{P}_{0,\infty}\text{MITL}_{0,\infty}$, which is known to be in PSPACE. We can repeat the same arguments for showing PSPACE-membership of checking the universality of the $S^*(\mathcal{A},\varphi)$ sets for a given PTA $\mathcal{A}$. Moreover, by Proposition 3 and Lemma 2, we get also PSPACE-membership of the emptiness problems for the V-sets in $\text{PMITL}_{\diamond}$. Therefore the following theorem holds.

Theorem 6. *The following problems are* PSPACE*-complete:*

1. *Checking the universality of each of the S-sets, restricted to input formulas from* $\text{PMITL}_{\Box}$.
2. *Checking the emptiness of each of the V-sets, restricted to input formulas from* $\text{PMITL}_{\diamond}$.

Table 1 summarizes the computational complexity of the considered decision problems for the logic PMITL and its fragments.

7. Decidable generalizations

In this section, we discuss two generalizations of the presented results. In the first one, we extend our logic by allowing *linear parametric expressions*.

Then we consider decision problems over the S-sets and the V-sets expressed as queries where each parameter is quantified either existentially or universally. In such formulation, the emptiness problem corresponds to a query where all the parameters are quantified existentially and the universality problem corresponds to one where all the parameters are quantified universally.

7.1. An extension of PMITL

In this section we discuss a syntactic extension of PMITL that we call PMITL_E and show that the considered decision problems are EXPSPACE-complete as for PMITL. We introduce first some notation.

A *linear expression* e is an expression of the form $c_0 + c_1x_1 + \ldots + c_mx_m$ with $c_0, c_1, \ldots, c_m \in \mathbb{Z}$ and $x_1, x_2, \ldots, x_m \in L \cup U$. We say that a parameter p_i *occurs positively* in e if $c_i > 0$ and *occurs negatively* in e if $c_i < 0$. We denote with $\mathcal{E}_u$ (resp. $\mathcal{E}_l$) the set of linear expressions over parameters in $U \cup L$ such that each parameter from U (resp. L) occurs positively and each parameter from L (resp. U) occurs negatively. Given a parameter valuation v and a *linear expression* e, with $v(e)$ we denote the integer $c_0 + c_1 v(x_1) + \ldots + c_m v(x_m)$, obtained by evaluating the parameters that occur in e by v. For an interval $I = (e, e')$, we denote with I_v the interval $(v(e), v(e'))$, that is the interval I where the expressions are replaced with their valuations.

We now introduce the logic PMITL_E. The syntax of PMITL_E formulas over the set of atomic propositions AP is defined by the following grammar:

$$\varphi := p \mid \neg p \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \varphi \mathcal{U}_H \varphi \mid \varphi \mathcal{R}_J \varphi$$

where $p \in AP$, and H and J are either defined as for PMITL or as follows:

- for $H = (a, b)$, $a \in \mathbb{N}$ and $b \in \mathcal{E}_u$;
- for $J = (a, b)$, $a \in \mathbb{N}$ and $b \in \mathcal{E}_l$.

Note that we have extended only the right endpoints of the intervals to *linear expression* e. The reason is that by allowing *linear expressions* also in the left endpoints the decision problems become undecidable (see Section 8). The semantics given for PMITL formulas directly applies to PMITL_E formulas (we have only changed the concept of parametric expressions and thus all the semantic changes are absorbed by the valuation of linear expressions).

To prove the satisfiability, validity and model-checking problems, we can repeat exactly the construction given in Section 5. In fact, all the lemmas on which the given proofs rely still hold when the formulas are from PMITL_E, and in all the results shown in that section, we can postpone handling the parametric linear expressions up to solving the decision problems for the Büchi L/U automaton corresponding to the starting formula (there is no need to valuate the intervals up to that point, and thus the parametric linear expressions are carried over through all the transformations). Recall that the parametric linear expressions are already allowed in the time constraints of the L/U automata given in [7], and that the recalled Theorem 1 has been shown for this more general definition of such automata. Therefore, we get the following theorem.

Theorem 7. *Checking the emptiness and the universality of S-sets is* Expspace*-complete, even when input formulas are from* PMITL_E.

7.2. A general decision problem over the sets of parameter valuations

In the previous sections, we have addressed the emptiness and universality queries for the sets $S(\varphi)$, $S(\mathcal{A},\varphi)$, $V(\varphi)$ and $V(\mathcal{A},\varphi)$. Here we define a general decision problem over such sets that includes the above emptiness and universality problems as instances.

Fix a PMITL formula φ and an arbitrary ordering $z_1,\ldots,z_n$ of the parameters used in φ. We say that an i-tuple of natural numbers, for $1 \le i \le n$, $(a_1,\ldots,a_i)$ is *admissible* if there is a valuation $v \in \mathcal{D}(\varphi)$ that assigns a_j to z_j, for all $j \in \{1,\ldots,i\}$.

Let $Q_1,\ldots,Q_n \in \{\forall,\exists\}$. We inductively define the sets $S_0,\ldots,S_n$ as follows. We let $S_n = S(\varphi)$ and for $i \in \{0,\ldots,n-1\}$, if $Q_{i+1} = \exists$, then let S_i be the set $\{(a_1,\ldots,a_i) \mid (a_1,\ldots,a_i,a_{i+1}) \in S_{i+1}$ for some natural number $a_{i+1}\}$; otherwise, i.e., $Q_{i+1} = \forall$, $S_i = \{(a_1,\ldots,a_i) \mid (a_1,\ldots,a_i,a_{i+1}) \in S_{i+1}$ for all the a_{i+1} such that $(a_1,\ldots,a_i,a_{i+1})$ is admissible$\}$.

In the following we denote the above set S_0 as $S(Q_1z_1\ldots Q_nz_n.\varphi)$.

Decision problem. The $Q_1z_1\ldots Q_nz_n$ *query over S* asks whether, given as input φ over the parameters $z_1,\ldots,z_n$, the set $S(Q_1z_1\ldots Q_nz_n).\varphi$ is not empty. Analogously, we define the $Q_1z_1\ldots Q_nz_n$ queries over the sets S^*, V and V^*.

Observe that the non-emptiness and universality problems considered in the previous sections correspond respectively to fully existential queries (i.e., each Q_i is $\exists$) and fully universal queries (i.e., each Q_i is $\forall$). Thus, by Theorem 3 we immediately have the following:

Remark 1. Let $\Gamma \in \{S, V, S^*, V^*\}$. If either $Q_1,\ldots,Q_n \in \{\exists\}$ or $Q_1,\ldots,Q_n \in \{\forall\}$, then deciding a $Q_1z_1\ldots Q_nz_n$ query over Γ is Expspace-complete.

Due to the polarity of the parameterized operators and the fact that the admissible values for each parameter are independent from the value assigned to the other parameters, in the introduced queries, the position of the existential quantifiers coupled with a parameter from L and of the universal quantifiers coupled with a parameter from U is not relevant. In fact, consider a query $Q_1z_1\ldots Q_nz_n$ over S. Fix a parameter z_i and denote with m_i its minimum admissible value. If $Q_i = \exists$ and $z_i \in L$, whenever $(a_1,\ldots,a_i,\ldots,a_n) \in S(\varphi)$, by Lemma 1 we have also $(a_1,\ldots,m_i,\ldots,a_n) \in S(\varphi)$, independently of the choice of a_j for each $j \ne i$. Thus, a query which existentially quantifies z_i is satisfied if and only if it is satisfied by assigning z_i with m_i, and therefore, we can move Q_iz_i to any other position of the sequence $Q_1z_1\ldots Q_nz_n$ without altering its validity. In the other case, i.e., $Q_i = \forall$ and $z_i \in U$, again by Lemma 1 and analogously to the previous case, we can argue that a query which universally quantifies z_i is satisfied if and only if is satisfied by assigning z_i with m_i. Therefore, we can move Q_iz_i to any other position of the sequence $Q_1z_1\ldots Q_nz_n$ without altering its validity.

The above observations are the main arguments to show the following lemma:

Lemma 11. *Let φ be a PMITL formula over the parameters $z_1,\ldots,z_n$ and $Q_1,\ldots,Q_n \in \{\forall,\exists\}$, and $\Gamma \in \{S, V, S^*, V^*\}$. For each $Q_1z_1\ldots Q_nz_n$ query over Γ there is an equivalent $Q_{i_1}z_{i_1}\ldots Q_{i_n}z_{i_n}$ query over Γ such that:*

- *$i_1,\ldots,i_n$ is a permutation of $1,\ldots,n$,*
- *$Q_{i_1},\ldots,Q_{i_k} \in \{\exists\}$ and $z_{i_1},\ldots,z_{i_k} \in L$, for some $k \in \{0,\ldots,n\}$,*
- *$Q_{i_{k+1}},\ldots,Q_{i_{k+j}} \in \{\forall\}$ and $z_{i_{k+1}},\ldots,z_{i_{k+j}} \in U$, for some $j \in \{0,\ldots,n-k\}$,*
- *for all $h \in \{k+j+1,\ldots,n\}$, $Q_{i_h} = \exists$ if and only if $z_{i_h} \in U$.*

Moreover, the sequence $i_1,\ldots,i_n$ can be effectively computed in linear time.

The above lemma suggests a simple algorithm to decide a $Q_1z_1\ldots Q_nz_n$ query when either all the parameters from U correspond to the quantifier $\forall$, or all the parameters from L correspond to the quantifier $\exists$. Suppose that the ordering of the parameters is as stated in Lemma 11. The algorithm starts by eliminating the parameters in subformulas of the forms $\Diamond_{(c+z,d)}\,\psi$ and $\Box_{(c+z,d)}\,\psi$, as in Theorem 3. Then, we assign all the parameters from L, in the scope of the quantifier $\exists$, and all the parameters from U, in the scope of the quantifier $\forall$, with the minimum admissible value in their domains. After this step, the remaining parameters are either all from U and in the scope of the quantifier $\exists$ or all from L and in the scope of the quantifier $\forall$. Thus, we use the decision algorithm as by Remark 1 on the resulting query. Therefore, we get the following theorem.

Theorem 8. *Let $z_1,\ldots,z_n$ be parameters, $Q_1,\ldots,Q_n \in \{\forall,\exists\}$, and $\Gamma \in \{S, V, S^*, V^*\}$. If either one of the following cases holds:*

- $Q_i = \exists$ *for each* $z_i \in L$, *or*
- $Q_i = \forall$ *for each* $z_i \in U$,

then deciding a query $Q_1 z_1 \dots Q_n z_n$ *over* Γ *is* EXPSPACE*-complete.*

Observe that the above theorem holds in particular when $z_1, \dots, z_n$ are all from L or all from U. Also, note that in general the above decision algorithm is not correct for arbitrary queries $Q_1 z_1 \dots Q_n z_n$ on arbitrary PMITL formulas and L/U automata. We are not aware of a solution for the general decision problem we have stated, and it is not clear to us whether the problem is even decidable.

8. Parameterization of time intervals

The need for restricting the use of each parameter with temporal operators of the same polarity has been already addressed in [2,17,7] for parametric temporal logics and parametric timed automata. The arguments used there also apply to PMITL and therefore we omit further discussion on this aspect.

In this section, we focus on the other restrictions we have placed on the definition of parameterized intervals of PMITL and PMITL_E. In particular, we relax the restriction that at most one of the endpoints of an interval is a parametric expression, and define three natural ways of adding parameters to both the endpoints of the intervals. Unfortunately, none of the proposed parameterizations leads to a decidable problems already for the logic PMITL. Moreover, we show that for PMITL_E also when only one of the interval endpoints can be parametric, but we allow parameterized intervals with parameters either in the left or the right endpoint, the considered problems for the resulting logic become undecidable.

For simplicity, in this section we consider only the satisfiability problem, that is the problem of checking the emptiness of the set $S(\varphi)$. The results for the other considered sets can be achieved similarly.

Parameterized time-shifts. In the first parameterization, we consider parameterized time-shifts of intervals. More precisely, with $\mathcal{L}_1$ we denote the logic obtained by augmenting MITL with parameterized intervals of the form $(c + x, d + x)$, such that (c, d) is not singular. Observe, that operators with this kind of intervals do not have polarity.

Theorem 9. *The problem of checking the emptiness of* $S(\varphi)$ *for any* φ *in* $\mathcal{L}_1$ *is undecidable. In particular, this holds already for the fragment of* $\mathcal{L}_1$ *with a single parameter x and where all parametric intervals are of the form* $(x, x+1)$.

Proof. We reduce the problem of checking the emptiness of sets $S(\varphi)$ for a formula φ of the logic PLTL (parametric LTL) augmented with equality in the parametric constraints coupled with the temporal operators. This problem is known to be undecidable even when a single parameter is allowed [2]. The logic PLTL is essentially the logic $\text{P}_{0,\infty}\text{MITL}_{0,\infty}$ augmented with the next-time operator, denoted $\bigcirc$, and with a discrete semantics (i.e., given with respect to infinite sequences of nodes labeled with atomic propositions).

The main idea is to capture a discrete-time sequence σ with a timed sequence α such that a position i in σ is captured by the interval $[i, i+1[$ within α. In this way, we can encode an equality constraint of the form $= x$ with the interval $[x, x+1[$ and the next-time operator with the operator $\Diamond_{[1,2[}$. Fix a formula φ from PLTL extended with equality and a single parameter x, and such parameter is only coupled with equality in the constraints. We translate φ into a formula φ' of $\mathcal{L}_1$ which uses only one parameter x. The formula φ' is the conjunction of two formulas. The first formula φ'_1 captures the timed sequences that encode the discrete sequences as described above. The second formula φ'_2 captures the requirements expressed by φ on such timed sequences.

We use a fresh atomic proposition p to denote the change of the discrete time (the "tick" of the discrete time clock). We require that $p \in \alpha(t)$ for all $t \in \mathbb{N}$ and $p \notin \alpha(t)$, otherwise. Thus the formula φ'_1 is:

$$p \wedge \Box_{[0,\infty[}\big(p \rightarrow (\neg p\,\mathcal{U}_{=1}\,p)\big) \wedge \bigwedge_{a \in AP} \Box_{[0,\infty[}\big((p \wedge a) \rightarrow \Box_{[0,1[}\,a\big),$$

where $\neg p\,\mathcal{U}_{=1}\,p$ is just an abbreviation for $(\neg p\,\mathcal{U}_{\leq 1}\,p) \wedge (\neg p\,\mathcal{U}_{\geq 1}\,p)$, and AP is the set of atomic proposition used in φ.

The formula φ'_2 is obtained from φ through the translation function τ defined inductively as follows, for any subformula ψ:

- if $\psi = a$ or $\psi = \neg a$ for $a \in AP$, then $\tau(\psi) = \psi$;
- for $\circ \in \{\wedge, \vee\}$, if $\psi = \psi_1 \circ \psi_2$, then $\tau(\psi) = \tau(\psi_1) \circ \tau(\psi_2)$;
- if $\psi = \bigcirc\psi'$, then $\tau(\psi) = \Diamond_{[1,2[}\,\tau(\psi')$;
- for $\nabla \in \{\mathcal{U}, \mathcal{R}\}$, if $\psi = \psi_1 \nabla_{\approx c} \psi_2$, then $\tau(\psi_1) \nabla_{\approx c} \tau(\psi_2)$ (where $c \in \mathbb{N}$ and $\approx \in \{\leq, <, >, \geq\}$);
- for $\nabla \in \{\mathcal{U}, \mathcal{R}\}$, if $\psi = \psi_1 \nabla_{=x} \psi_2$, then $\tau(\psi_1) \nabla_{[x,x+1[}\,\tau(\psi_2)$.

Denoting $\psi'_2 = \tau(\varphi)$, we have that $S(\varphi)$ is empty if and only if $S(\varphi'_1 \wedge \varphi'_2)$ is empty, and thus the theorem holds. $\Box$

Full parameterization. We extend PMITL with parameterized intervals where both left endpoints and right endpoints are in $\mathcal{E}(L) \cup \mathcal{E}(U)$. More precisely, given $x \in U$ and $y \in L$, we consider *fully parameterized* intervals which can be of the form $(c + y, d + x)$, when used as subscripts of *until* operators, and of the form $(c + x, d + y)$, when used as subscripts of *release* operators. We denote this logic $\mathcal{L}_2$.

Theorem 10. *The problem of checking the emptiness of $S(\varphi)$ for any φ in $\mathcal{L}_2$ is undecidable.*

Proof. Consider the formula $\varphi = \Diamond_{[y,x+1]} \psi \wedge \Box_{[x+1,y+2]} true$. For an admissible parameter valuation v, it holds that $v(y) < v(x) + 1$ and $v(x) + 1 < v(y) + 2$ from which we obtain that $v(x) = v(y)$. Thus, φ is equivalent to the formula $\Diamond_{[z,z+1]} \psi$ of $\mathcal{L}_1$. Therefore, the theorem follows from Theorem 9. $\square$

Another way of obtaining full parametrization of the intervals is to use a parameter for translating the interval in time and the other to adjust the width of the interval. Let $\mathcal{L}_3$ denote the corresponding logic. We can show the undecidability by using the following reduction. Given an interval $(c + y, d' + y + x)$, we obtain the interval $(c + y', d + x')$ by the linear transformation: $y' = y$, $x' = c + 1 + x + y' - d$, and $d' = c + 1$. Thus, from Theorem 10 we get:

Theorem 11. *The problem of checking the emptiness of $S(\varphi)$ for any φ in $\mathcal{L}_3$ is undecidable.*

Parameters as left endpoints in PMITL$_E$. We have defined PMITL$_E$ such that the parametric linear expressions can be used only as right endpoints of the parameterized intervals. Here we relax this restriction and admits such expressions also as left endpoints of the intervals, but we keep the restriction that only one endpoint is parameterized. We call $\mathcal{L}_4$ the resulting logic. The following theorem holds.

Theorem 12. *The problem of checking the emptiness of $S(\varphi)$ for any φ in $\mathcal{L}_4$ is undecidable.*

Proof. Consider the formula $\varphi = \Box_{[x-y,1]} true \wedge \neg\psi\mathcal{U}_{]0,x]}\psi \wedge \neg\psi\mathcal{U}_{[y,\infty[}\psi$. For an admissible parameter valuation v, it holds that $0 \leq v(x) - v(y) < 1$, from which we obtain that $v(x) = v(y)$. Thus, φ is equivalent to the formula $\neg\psi\mathcal{U}_{=x}\psi$. It is known that augmenting PLTL with such kind of formulas leads to undecidability [2]. We can use a translation as in the proof of Theorem 9 to reduce the emptiness problem for the sets $S(\varphi)$ in this discrete-time logic to show undecidability of the same problem in $\mathcal{L}_4$. $\square$

9. Conclusion

In this paper we have addressed several aspects of introducing parametric constants to express timing constraints in real-time linear-time temporal logic. In particular, we have proposed the logic PMITL as a parametric extension of MITL and considered problems that are the analogous of, and extend to PMITL, the satisfiability, validity and model-checking problems for MITL. We show that under some restrictions on the use of parameters such problems turn out to be decidable and Expspace-complete (same as for the MITL decision problems). We also show that the considered restrictions are needed, in fact, if we relax any of them the decision problems become undecidable.

It is useful to be able to characterize the domains of the parameter valuations that make the considered analysis true. Unfortunately, this is possible only when we restrict to use only parameterized operators of just one polarity. The reason is essentially that if we can characterize such domains in an algorithmically usable format for the full logic, we would be able to answer the decision problems also when parameter valuations evaluating parameterized intervals to singular sets are allowed. This was first shown for the discrete-time logic PLTL [2], then for Büchi L/U automata [7], and we can replicate the positive results shown there also for our logic.

This impossibility of characterizing the space of the fulfilling parameter valuations has motivated the formulation of a general decision problem which gives more information on this space than simply considering the emptiness and universality problems. We have considered general queries on such spaces where each parameter is either quantified existentially or universally. We have solved this query decision problem with the restriction that either all the lower bound parameters are existentially quantified or all the upper bound parameters are existentially quantified. Note that even if this does not cover all the cases, it gives us a feedback on the fulfilling valuations of formulas with both kinds of parameters, and thus covers cases where the characterization of this space does not look possible. We leave open the problem when general queries are allowed. The problem of the generalization of these results when parametric linear expressions are allowed as right endpoints of the intervals is also open. In this case, the main problem seems to be the fact that the admissible values for a parameter are depending on the values assigned to the other parameters, and therefore, it looks hard to show that the inversion of quantifiers does not alter the meaning of a query.

Besides the above open problems, another interesting future research direction is to explicitly constrain the space of the admissible valuations with Boolean combinations of linear systems over the parameters. Such a study has been already addressed for Büchi L/U automata in [7]. Concerning the universality and emptiness problems, we are able to show the same results as in [7] within our formalism. However, it is not clear to us if the extension of these results to the query decision problem is feasible.

Also, the expressiveness of the fragments of PMITL we have considered requires further investigation. In particular, our results show that the fragments $\mathrm{P}_{0,\infty}\mathrm{MITL}$ and $\mathrm{PMITL}_{0,\infty}$ are both equivalent to PMITL (with $\mathrm{P}_{0,\infty}\mathrm{MITL}$ we denote the fragment of PMITL where all the parameterized intervals are as in $\mathrm{P}_{0,\infty}\mathrm{MITL}_{0,\infty}$). However we do not know if $\mathrm{P}_{0,\infty}\mathrm{MITL}_{0,\infty}$ can express the whole PMITL. For the non-parameterized analogues of these logics, i.e., $\mathrm{MITL}_{0,\infty}$ and MITL, equivalence holds though MITL is exponentially more succinct than $\mathrm{MITL}_{0,\infty}$ [3]. The translation given there uses negation, and therefore, it does not seem to be suitable for obtaining a similar translation in the parameterized settings.

Further, here we have considered only integer parameter valuations. For many practical uses this is the case, however our results can be directly used to answer the considered decision problems (universality and emptiness of the set of fulfilling valuations) also when real-valued parameters are allowed. Though, in this extension, the characterization of the set of the fulfilling valuations when only parameters of one polarity are used deserves further investigation. In fact, by our results we are only able to approximate such sets starting from their minimal/maximal integer parameter valuations (by exploiting the fact that they are upward/downward closed).

Finally, it would be interesting to investigate parametric extensions of timed games. Finite game graphs with winning conditions expressed as PLTL formulas have been recently solved in [27]. As for timed games, the approach of [18] has been recently extended to parametric timed reachability games in [19]. Towards the study of parametric timed games with PMITL formulas, we believe that it would be useful to extend to the parametric setting the automata-theoretic approach to solve timed games with temporal logic winning conditions given in [16] and timed tree automata (see [22,23]).

References

[1] R. Alur, C. Courcoubetis, D.L. Dill, Model-checking in dense real-time, Inform. Comput. 104 (1) (1993) 2–34.
[2] R. Alur, K. Etessami, S. La Torre, D. Peled, Parametric temporal logic for “model measuring”, ACM Trans. Comput. Log. 2 (3) (2001) 388–407.
[3] R. Alur, T. Feder, T.A. Henzinger, The benefits of relaxing punctuality, J. ACM 43 (1) (1996) 116–146.
[4] R. Alur, T.A. Henzinger, M.Y. Vardi, Parametric real-time reasoning, in: STOC, 1993, pp. 592–601.
[5] É. André, T. Chatain, L. Fribourg, E. Encrenaz, An inverse method for parametric timed automata, Internat. J. Found. Comput. Sci. 20 (5) (2009) 819–836.
[6] E. Asarin, A. Donzé, O. Maler, D. Nickovic, Parametric identification of temporal properties, in: S. Khurshid, K. Sen (Eds.), RV, in: Lecture Notes in Computer Science, vol. 7186, Springer, 2011, pp. 147–160.
[7] L. Bozzelli, S. La Torre, Decision problems for lower/upper bound parametric timed automata, Formal Methods Syst. Des. 35 (2) (2009) 121–151.
[8] V. Bruyère, E. Dall'Olio, J.-F. Raskin, Durations and parametric model-checking in timed automata, ACM Trans. Comput. Log. 9 (2) (2008).
[9] V. Bruyère, J.-F. Raskin, Real-time model-checking: parameters everywhere, Log. Methods Comput. Sci. 3 (1) (2007).
[10] D. Bundala, J. Ouaknine, Advances in parametric real-time reasoning, in: Erzsébet Csuhaj-Varjú, Martin Dietzfelbinger, Zoltán Ésik (Eds.), MFCS, in: Lecture Notes in Computer Science, vol. 8634, Springer, 2014, pp. 123–134.
[11] S.V.A. Campos, E.M. Clarke, O. Grumberg, Selective quantitative analysis and interval model checking: verifying different facets of a system, in: R. Alur, T.A. Henzinger (Eds.), CAV, in: Lecture Notes in Computer Science, vol. 1102, Springer, 1996, pp. 257–268.
[12] R. Chevallier, E. Encrenaz-Tiphène, L. Fribourg, W. Xu, Timed verification of the generic architecture of a memory circuit using parametric timed automata, Formal Methods Syst. Des. 34 (1) (2009) 59–81.
[13] C. Courcoubetis, M. Yannakakis, Minimum and maximum delay problems in real-time systems, Formal Methods Syst. Des. 1 (4) (1992) 385–415.
[14] B. Di Giampaolo, S. La Torre, M. Napoli, Parametric metric interval temporal logic, in: A.H. Dediu, H. Fernau, C. Martín-Vide (Eds.), LATA, in: Lecture Notes in Computer Science, vol. 6031, Springer, 2010, pp. 249–260.
[15] E.A. Emerson, R.J. Trefler, Parametric quantitative temporal reasoning, in: LICS, 1999, pp. 336–343.
[16] M. Faella, S. La Torre, A. Murano, Automata-theoretic decision of timed games, in: A. Cortesi (Ed.), VMCAI, in: Lecture Notes in Computer Science, vol. 2294, Springer, 2002, pp. 94–108.
[17] T. Hune, J. Romijn, M. Stoelinga, F.W. Vaandrager, Linear parametric model checking of timed automata, J. Log. Algebr. Program. 52–53 (2002) 183–220.
[18] A. Jovanovic, D. Lime, O.H. Roux, Integer parameter synthesis for timed automata, in: N. Piterman, S.A. Smolka (Eds.), TACAS, in: Lecture Notes in Computer Science, vol. 7795, Springer, 2013, pp. 401–415.
[19] A. Jovanovic, D. Lime, O.H. Roux, Synthesis of bounded integer parameters for parametric timed reachability games, in: D.V. Hung, M. Ogawa (Eds.), ATVA, in: Lecture Notes in Computer Science, vol. 8172, Springer, 2013, pp. 87–101.
[20] R. Koymans, Specifying real-time properties with metric temporal logic, Real-Time Syst. 2 (4) (1990) 255–299.
[21] O. Kupferman, N. Piterman, M.Y. Vardi, From liveness to promptness, Formal Methods Syst. Des. 34 (2) (2009) 83–103.
[22] S. La Torre, M. Napoli, Timed tree automata with an application to temporal logic, Acta Inform. 38 (2) (2001) 89–116.
[23] S. La Torre, M. Napoli, Finite automata on timed omega-trees, Theoret. Comput. Sci. 293 (3) (2003) 479–505.
[24] A. Pnueli, The temporal logic of programs, in: FOCS, IEEE, 1977, pp. 46–57.
[25] F. Wang, Parametric timing analysis for real-time systems, Inform. Comput. 130 (2) (1996) 131–150.
[26] H. Yang, B. Hoxha, G.E. Fainekos, Querying parametric temporal logic properties on embedded systems, in: B. Nielsen, C. Weise (Eds.), ICTSS, in: Lecture Notes in Computer Science, vol. 7641, Springer, 2012, pp. 136–151.
[27] M. Zimmermann, Optimal bounds in parametric LTL games, Theoret. Comput. Sci. 493 (2013) 30–45.