Journal of Universal Computer Science, vol. 15, no. 17 (2009), 3273-3304
submitted: 15/11/08, accepted: 15/10/09, appeared: 1/11/09 © J.UCS

Parametric Model-Checking of Stopwatch Petri Nets

Louis-Marie Traonouez
(IRCCyN, Nantes, France
Louis-Marie.Traonouez@irccyn.ec-nantes.fr)

Didier Lime
(IRCCyN, Nantes, France
Didier.Lime@irccyn.ec-nantes.fr)

Olivier (H.) Roux
(IRCCyN, Nantes, France
Olivier-h.Roux@irccyn.ec-nantes.fr)

Abstract: At the border between control and verification, parametric verification can be used to synthesize constraints on the parameters to ensure that a system verifies given specifications. In this paper we propose a new framework for the parametric verification of time Petri nets with stopwatches. We first introduce a parametric extension of time Petri nets with inhibitor arcs (ITPNs) with temporal parameters and we define a symbolic representation of the parametric state-space based on the classical state-class graph method. Then, we propose semi-algorithms for the parametric model-checking of a subset of parametric TCTL formulae on ITPNs. These results have been implemented in the tool Romeo and we illustrate them in a case-study based on a scheduling problem.

Key Words: time Petri nets, stopwatches, model-checking, parameters, state-class graph
Category: D.2.2, D.2.4

1 Introduction

In the class of timed discrete event systems, timed extensions of Petri nets (see [Bowden, 1996] for a survey) and timed automata (TA) [Henzinger et al., 1994] are widely used to model and analyze such concurrent systems. Time Petri nets [Merlin, 1974] allow an easy representation of real-time systems features such as synchronization and parallelism. State reachability is decidable for bounded *TPNs*, which is sufficient for virtually all practical purposes.

However, modeling of many embedded systems requires to express suspension and resumption of actions. This implies extending traditional clock variables by "stopwatches". Several extensions of TPNs address this issue by controlling stopwatches either with priorities [Bucci et al., 2004, Roux and Déplanche, 2002] or inhibitor arcs [Roux and Lime, 2004] or activator arcs [Berthomieu et al., 2007]. These models all belong to the class of *TPNs* extended with stopwatches (*SwPNs*), for which the reachability problem is undecidable [Berthomieu et al., 2007].

Currently, the use of real-time systems is quickly increasing and, at the same time, correctness proofs on these systems must be provided. Formal methods such as model-checking allow the verification of a system by exploring the state-space of a model. The model-checking of timed models has become more and more efficient. It nevertheless requires a complete knowledge of the system. Consequently, the verification of the behavior can be performed only after the design stage when the global system and its environment are known. On the one hand, it increases the complexity of the conception and the verification of systems. For too complex systems this can lead to a combinatorial explosion. Besides, if the system is proven wrong or if the environment changes, this complex verification process must be carried out again. On the other hand, getting a complete knowledge of a system can be impossible. In many important applications, a system is defined by parameters that are in relation with several other systems. In the existing tools for modelling and verification, parameters are often used, however they must be instantiated to perform analyses. The next development step of the technology is to be able to directly analyze a parametric model.

1.1 Control vs. Verification

The *verification* problem for a given system S and a specification ϕ consists in checking whether S satisfies ϕ which is often written $S \models \phi$ and referred to as the *model-checking problem*. The *control problem* assumes the system is *open i.e.* we can restrict the behavior of S: some events in S are *controllable* and the others are *uncontrollable*, and we can sometimes disable controllable actions. The *control problem* for a system S and a specification ϕ asks the following: Is there a controller C s.t. $S \times C \models \phi$? The associated *control synthesis problem* asks to compute a witness controller C.

The *parametric model-checking problem* lies at the interface between the previous two problems. For a parametric system S and a parametric specification ϕ, it consists in checking whether there exists a valuation ν of the parameters such that S satisfies ϕ for this valuation, which is written $[\![S]\!]_\nu \models [\![\phi]\!]_\nu$. The associated *parametric synthesis problem* computes the set of valuations Γ such that $\forall \nu \in \Gamma$, $[\![S]\!]_\nu \models [\![\phi]\!]_\nu$.

1.2 Related Works

Parametric analysis of real-time systems has been studied in [Alur et al., 1993]. They introduce Parametric Timed Automata (PTA) and prove that, in the general case, the emptiness problem is undecidable. In [Hune et al., 2001] the authors prove that for a particular class of PTA called L/U automata this emptiness problem is decidable. They also give a model-checking algorithm that uses parametric Difference Bound Matrices. Parametric model-checking can be used to generate a set of constraints on the parameters such that the property is verified. In discrete time, parametric model-checking of PTA has been studied in [Bruyère and Raskin, 2007], and some decidability results have been found. On hybrid automata, state-space exploration algorithms have been extended to allow a parametric analysis and implemented in the tool HYTECH [Henzinger et al., 1997].

Another approach developed in [Wang, 1996] focuses on the verification of parametric TCTL formulae on clock automata. They consider unbounded parameters that take their value among integers and the problem is proven decidable. In [Virbitskaite and Pokozy, 1999], this approach is used in parametric *TPNs*, but with bounded parameters. However, they consider and analyze a region graph for each parameter valuation.

1.3 Our Contribution

This paper is an extended version of [Traonouez et al., 2008] that includes detailed proofs and presents an implementation used to analyze a case-study.

We propose to study the *parametric synthesis problem* on stopwatch Petri nets. The developments in this paper will consider time Petri nets with inhibitor arcs (*ITPNs*) but they can be applied to the others models of stopwatch Petri nets. *ITPNs* are extended with time parameters that can be used in the firing intervals of the transitions.

We consider unbounded parameters and thus, we need a proper abstraction of the state-space of the parametric model. In *TPNs*, considering that the time is dense, the state-space of the model is infinite, but it can be represented by a finite partition as in the state-class graph [Berthomieu and Diaz, 1991]. We therefore extend the state-class graph construction with parameters and define parametric state-classes that represent at the same time the temporal domain and the parameter domain.

Although the state-class graph does not preserve timed properties, there exists methods [Hadjidj and Boucheneb, 2006] to verify a subset of TCTL with this abstraction. We consider this subset of formulae and extend it with parameters to define parametric TCTL formulae. Then, we propose and prove semi-algorithms to solve the *parametric synthesis problem*.

1.4 Outline of the Paper

In section 2, we present our parametric extension of *ITPNs* (*PITPNs*). Then, in section 3, we introduce some decidability and undecidability results. Section 4 defines the parametric state-class graph of *PITPNs*. In section 5, we study the parametric model-checking of a subset of TCTL with parameters. In section 6, we discuss our solution to the parametric model-checking problem and we present its implementation in the tool ROMEO. Finally, in section 7 our method is applied to a case-study of a scheduling problem.

2 Parametric Time Petri Nets with Inhibitor Arcs

2.1 Notations

The sets $\mathbb{N}$, $\mathbb{Q}^+$ and $\mathbb{R}^+$ are respectively the sets of natural, non-negative rational and non-negative real numbers. An interval I of $\mathbb{R}^+$ is a $\mathbb{Q}$-interval iff its left endpoint ${}^{\uparrow}I$ belongs to $\mathbb{Q}^+$ and its right endpoint $I^{\downarrow}$ belongs to $\mathbb{Q}^+ \cup \{\infty\}$. We denote by $\mathcal{I}(\mathbb{Q})$ the set of $\mathbb{Q}$-intervals of $\mathbb{R}^+$.

2.2 Formal Definitions of PITPNs

We parameterize the *ITPN* model with a set of temporal parameters $Par = \{\lambda_1, \lambda_2, \ldots, \lambda_l\}$ by replacing some of the temporal bounds of the transitions by parameters. These parameters are considered as constant variables in the semantics, and take their values among rationals.

Some initial constraints are given on the parameters. These constraints define the domain $D_p \subseteq {\mathbb{Q}^+}^{Par}$ of the parameters which is a convex polyhedron. These constraints must at least specify that for all parameters valuations in D_p, the minimum bounds of the firing intervals of the transitions are inferior to the maximum bounds. Additional linear constraints may of course be given.

A *valuation* of the parameters is a function $\nu : Par \rightarrow \mathbb{Q}^+$, such that $[\nu(\lambda_1)\ \nu(\lambda_2) \ldots \nu(\lambda_l)]^\top \in D_p$, which is equivalent to say that ν is a point of D_p. We will also write that $\nu = [\lambda_1\ \lambda_2 \ldots \lambda_l]^\top$.

A *linear constraint* over the parameters is an expression $\gamma = \sum_{i=0}^{l} a_i * \lambda_i \sim b$, where $\forall 0 \leq i \leq l,\ a_i, b \in \mathbb{Q}$ and $\sim \in \{=, <, >, \leq, \geq\}$. A convex polyhedron is a conjunction of linear constraints. We write that $\lambda_i \in \gamma$ iff $a_i \neq 0$.

A *parametric time interval* is a function $J : D_p \rightarrow \mathcal{I}(\mathbb{Q}^+)$ that associates to each parameter valuation a $\mathbb{Q}$-interval. The set of parametric time intervals over Par is denoted by $\mathcal{J}(Par)$. As for numerical interval, J can be split into two functions ${}^{\uparrow}J$ and $J^{\downarrow}$ that are respectively the minimum bound and the maximum bound. They can be both represented by a linear constraint over the parameters.

Definition 1. A parametric time Petri net with inhibitor arcs (*PITPN*) is a tuple $\mathcal{N} = \langle P, T, Par, {}^{\bullet}(.), (.)^{\bullet}, {}^{\circ}(.), M_0, J_s, D_p \rangle$, where:

- $P = \{p_1, p_2, \ldots, p_m\}$ is a non-empty finite set of *places*,
- $T = \{t_1, t_2, \ldots, t_n\}$ is a non-empty finite set of *transitions*,
- $Par = \{\lambda_1, \lambda_2, \ldots, \lambda_l\}$ is a finite set of *parameters*,

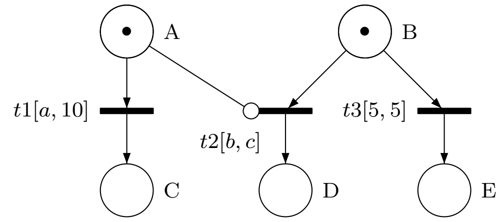

Figure 1: A parametric time Petri net with inhibitor arcs

- ${}^\bullet(.) \in (\mathbb{N}^P)^T$ is the *backward incidence function*,
- $(.)^\bullet \in (\mathbb{N}^P)^T$ is the *forward incidence function*,
- ${}^\circ(.) \in (\mathbb{N}^P)^T$ is the *inhibition function*,
- $M_0 \in \mathbb{N}^P$ is the *initial marking* of the net,
- $J_s \in (\mathcal{J}(Par))^T$ is the function that associates a *parametric firing interval* to each transition,
- $D_p \subseteq {\mathbb{Q}^+}^{Par}$ is a convex polyhedron that is the *domain of the parameters*.

A *marking* M of the net is an element of $\mathbb{N}^P$ such that $\forall p \in P, M(p)$ is the number of tokens in the place p.

A transition t is said to be *enabled* by the marking M if $M \geq^\bullet t$, (*i.e.* if the number of tokens in M in each input place of t is greater or equal to the value on the arc between this place and the transition). We denote it by $t \in \mathsf{enabled}\,(M)$.

A transition t is said to be *inhibited* by the marking M if the place connected to one of its inhibitor arc is marked with at least as many tokens than the weight of the considered inhibitor arc between this place and t: $0 < {}^\circ t \leq M$. We denote it by $t \in \mathsf{inhibited}\,(M)$. Practically, inhibitor arcs are used to stop the elapsing of time for some transitions: an inhibitor arc between a place p and a transition t means that the stopwatch associated to t is stopped as long as place p is marked with enough tokens.

A transition t is said to be *active* in the marking M if it is enabled and not inhibited by M.

Example 1. In the figure 1 an example of *PITPN* is given that includes three parameters a, b and c, and an inhibitor arc from place A to transition t_2. The domain of parameters is defined by:

$$D_p = \begin{cases} 0 \leq a \leq 10, \\ 0 \leq b \leq c. \end{cases}$$

2.3 Semantics of Parametric Time Petri Nets with Inhibitor Arcs

The semantics of a Parametric Time Petri net with Inhibitor Arcs $\mathcal{N}$ is defined for a parameter valuation $\nu \in D_p$, as the non-parametric *ITPN* obtained when replacing in $\mathcal{N}$ all the parameters by their valuation.

Definition 2 Semantics of a PITPN. Given a *PITPN* $\mathcal{N} = \langle P, T, Par, {}^\bullet(.), (.)^\bullet, {}^\circ(.), M_0, J_s, D_p \rangle$, and a valuation $\nu \in D_p$, the semantics $[\![\mathcal{N}]\!]_\nu = \langle P, T, {}^\bullet(.), (.)^\bullet, {}^\circ(.), M_0, I_s \rangle$ of $\mathcal{N}$ is an *ITPN* such that $\forall t \in T,\ I_s(t) = J_s(t)(\nu)$.

We now recall the semantics of an *ITPN*.

A transition t is said to be *firable* when it has been enabled and not inhibited for at least ${}^\uparrow I_s(t)$ time units.

A transition t_k is said to be *newly* enabled by the firing of the transition t_i from the marking M, and we denote it by $\uparrow \mathsf{enabled}\,(t_k, M, t_i)$, if the transition is enabled by the new marking $M - {}^\bullet t_i + t_i^\bullet$ but was not by $M - {}^\bullet t_i$, where M is the marking of the net before the firing of t_i. Formally:

$$\begin{aligned}&\uparrow \mathsf{enabled}\,(t_k, M, t_i) = ({}^\bullet t_k \leq M - {}^\bullet t_i + t_i^\bullet)\\ &\wedge((t_k = t_i) \vee ({}^\bullet t_k > M - {}^\bullet t_i))\end{aligned}$$

By extension, we will denote by $\uparrow \mathsf{enabled}\,(M, t_i)$ the set of transitions newly enabled by firing the transition t_i from the marking M.

Definition 3. A *state* of an *ITPN* is a pair $q = (M, I)$ in which M is a marking and I is a function called the *interval* function. Function $I \in (\mathcal{I}(\mathbb{Q}))^T$ associates a temporal interval with every transition enabled at M.

The semantics of an *ITPN* is defined as a timed transition system (TTS) [Larsen et al., 1995], in which two kinds of transitions may occur: *time* transitions when time passes and *discrete* transitions when a transition of the net is fired.

Definition 4 Semantics of an ITPN. The semantics of a time Petri net with inhibitor arcs $\mathcal{N} = \langle P, T, {}^\bullet(.), (.)^\bullet, {}^\circ(.), M_0, I_s \rangle$ is defined as the TTS $\mathcal{S}_\mathcal{N} = \langle Q, q_0, \rightarrow \rangle$ such that:

- $Q = \mathbb{N}^P \times (\mathcal{I}(\mathbb{Q}))^T$,
- $q_0 = (M_0, I_s)$,
- $\rightarrow \in Q \times (T \cup \mathbb{R}^+) \times Q$ is the transition relation including a time transition relation and a discrete transition relation.

The time transition relation is defined $\forall d \in \mathbb{R}^+$ by:

$$(M, I) \xrightarrow{d} (M, I') \text{ iff } \forall t_i \in T,$$
$$\begin{cases} I'(t_i) = \begin{cases} I(t_i) \text{ if } t_i \in \mathsf{enabled}\,(M) \text{ and } t_i \in \mathsf{inhibited}\,(M) \\ {}^{\uparrow}I'(t_i) = \max(0, {}^{\uparrow}I(t_i) - d), \text{ and } I'(t_i)^{\downarrow} = I(t_i)^{\downarrow} - d \text{ otherwise,} \end{cases} \\ M \geq {}^{\bullet}t_i \Rightarrow I'(t_i)^{\downarrow} \geq 0 \end{cases}$$

The discrete transition relation is defined $\forall t_i \in T$ by:

$$(M, I) \xrightarrow{t_i} (M', I') \text{ iff } \begin{cases} t_i \in \mathsf{enabled}\,(M) \text{ and } t_i \notin \mathsf{inhibited}\,(M), \\ M' = M - {}^{\bullet}t_i + t_i^{\bullet}, \\ {}^{\uparrow}I(t_i) = 0, \\ \forall t_k \in T, I'(t_k) = \begin{cases} I_s(t_k) \text{ if } \uparrow\mathsf{enabled}\,(t_k, M, t_i) \\ I(t_k)\ otherwise \end{cases} \end{cases}$$

Example 2. For the *PITPN* presented in the figure 1, if we consider the valuation $\nu = (6, 2, 3)$ of the three parameters (a, b, c), then in the semantics of $[\![\mathcal{N}]\!]_\nu$:

- transition t_1 is enabled but not firable (because t_3 must be fired before it),
- transition t_2 is inhibited by the token in place A,
- the only firable transition is t_3 at date 5.

A *run* ρ of length $n \geq 0$ in $\mathcal{S}_\mathcal{N}$ is a finite or infinite sequence of alternating time and discrete transitions of the form

$$\rho = q^0 \xrightarrow{d_0} q^0 + d_0 \xrightarrow{t_0} q^1 \xrightarrow{d_1} q^1 + d_1 \xrightarrow{t_1} \cdots q^n \xrightarrow{d_n} \cdots$$

We write *first*(ρ) the first state of a run ρ. A run is *initial* if *first*$(\rho) = q_0$. A run ρ of $\mathcal{N}$ is an initial run of $\mathcal{S}_\mathcal{N}$. For a state q, the set of all the infinite runs starting from q is denoted by $\pi(q)$. The set of all the runs of $\mathcal{N}$ is $\pi(q_0)$.

For a state q^i in ρ the absolute time elapsed (relative to the initial state) is $\mathsf{time}(q) = d_0 + d_1 + \cdots + d_{i-1}$. For a run ρ the total elapsed time in ρ is $\mathsf{time}(\rho) = \sum_{i=0}^{n} d_i$. In this paper we restrict ourselves to non-zeno *ITPN*, which means that the elapsed time is diverging (i.e. $\forall \rho \in \pi(q_0), \mathsf{time}(\rho) = \infty$), and by extension to non-zeno *PITPN* (i.e. such that $\forall \nu \in D_p$, $[\![\mathcal{N}]\!]_\nu$ is non zeno).

3 Decidability of Parametric TPNs

In this section, we give some results concerning the decidability of the emptiness and reachability problems for bounded parametric time Petri nets (without inhibitor arcs). The case of *PITPNs* is of little interest since these problems are already known undecidable for bounded *ITPNs* [Berthomieu et al., 2007].

Let us consider lower/upper bound (L/U) bounded parametric *TPNs* i.e. every parameter occurring in the *PTPN* is either in the lower bound of some of the parametric time intervals or in their upper bound, but there exists no pair of parametric intervals J_1, J_2 and no parameter λ such that $\lambda \in {}^{\uparrow}I_1$ and $\lambda \in I_2^{\downarrow}$.

Theorem 5. *The emptiness and reachability problems for bounded L/U parametric TPNs are decidable.*

Proof. The structural and syntactical translation proposed in [Cassez et al., 2006] from a *TPN* into a bisimilar timed automaton (*TA*) can straightforwardly be extended from L/U *PTPNs* to L/U parametric *TA* [Hune et al., 2001]. Therefore, since the emptiness and reachability problems are decidable for L/U parametric *TA* [Hune et al., 2001], they also are decidable for L/U *PTPNs*.

Theorem 6. *The emptiness and reachability problems for bounded parametric TPNs are undecidable.*

Proof. The structural and syntactical translation preserving timed language acceptance proposed in [Bérard et al., 2005] from a *TA* into a bounded *TPN* can straightforwardly be extended to parametric *TA*. Thus, for every parametric *TA*, we can compute a parametric *TPN* that accepts the same timed language. Since the emptiness problem (and then, the reachability problem) is undecidable for parametric *TA* [Alur et al., 1993], it is also undecidable for parametric *TPNs*.

4 The Parametric State-Class Graph of a PITPN

Since the state-space of a non-parametric *TPN* is generally infinite in dense-time, it is required to abstract the time by merging some states into some equivalence classes. Consequently, symbolic representations of the state-space are used. One of the approaches to partition the state-space in a finite set of infinite state classes is the state-class graph [Berthomieu and Diaz, 1991]. This approach has been extended for *ITPNs* in [Roux and Lime, 2004].

However, there also exists an infinite number of parameters valuations. Thus, in the same way, we need to use symbolic representations of the parameters domains. In time Petri nets or timed automata, the time domain of an abstract state can be efficiently encoded by a difference bound matrix (DBM). This is why, in the parametric timed automata proposed in [Hune et al., 2001], the authors define parametric DBMs in which they encode both the time domains and the parameters domains. When considering stopwatch time Petri nets, the firing domain of a class is a general polyhedron and cannot necessarily be represented by a DBM. Consequently, in the parametric state-classes of *PITPNs* we will use polyhedra to encode both the transitions variables domains and the parameters domains.

4.1 Parametric State-Classes

Definition 7. A *parametric state-class* C of a *PITPN* is a pair (M, D) where M is a marking of the net and D is a firing domain represented by a (convex) polyhedron involving $l + n$ variables, with n being the number of transitions enabled by the marking of the class and l the number of parameters in the net.

A point $(\nu|\nu')$ of the firing domain is constituted by a valuation ν of the parameters in Par and a valuation ν' of the firing times θ of enabled transitions. The set of those variables θ of D will be noted Θ.

We denote by $D_{|Par}$ the projection of a firing domain D on the set of parameters:

$$D_{|Par} = \{\nu \in {\mathbb{Q}^+}^l \mid \exists \nu' \in \mathbb{R}^n \text{ s.t. } (\nu|\nu') \in D\}$$

This definition can be extended to any arbitrary subset of variables of D.

4.2 Computation of the Parametric State-Class Graph

The parametric state-class graph is computed similarly to the non-parametric case. Parameters are embedded into the firing domain of the initial class, and the operations that compute the successor classes do not concern the parameters. However, throughout the computation of the graph, the domain of the parameters in a class will be automatically reduced to consider only the valuations that make this class reachable.

Definition 8 Firability. Let $C = (M, D)$ be a parametric state-class of a *PITPN*. A transition t_i is said to be *firable* from C iff there exists a solution $(\nu|\nu') \in D$, such that $\forall j \in \{1, \ldots, n\} - \{i\}$, s.t. $t_j \in \mathsf{enabled}(M)$ and $t_j \notin \mathsf{inhibited}(M)$, $\nu'(\theta_i) \leq \nu'(\theta_j)$. We will write this: $t_i \in \mathsf{firable}(C)$.
It means that for the valuation ν there exists a state in C in which t_i is *firable*.

Now, given a parametric class $C = (M, D)$ and a firable transition t_f, the parametric class $C' = (M', D')$ obtained from C by the firing of t_f, which we write $C' = succ(C, t_f)$, is given by:

- $M' = M - {}^\bullet t_f + t_f^\bullet$
- D' is computed along the following steps, and noted $\mathsf{next}(D, t_f)$
  1. intersection with the firability constraints : $\forall j$ s.t. t_j is active, $\theta_f \leq \theta_j$
  2. variable substitutions for all enabled transitions that are *active* t_j: $\theta_j = \theta_f + \theta'_j$,
  3. elimination (using for instance the Fourier-Motzkin method) of all variables relative to transitions disabled by the firing of t_f,

4. addition of inequations relative to newly enabled transitions

$$\forall t_k \in \uparrow \mathsf{enabled}\,(M, t_f)\,,\ {}^{\uparrow}J_s(t_k) \leq \theta'_k \leq J_s(t_k)^{\downarrow}$$

The variable substitutions correspond to a shift of time origin for active transitions: the new time origin is the firing time of t_f. t_f is supposed to be firable so the polyhedron constrained by the inequalities $\theta_f \leq \theta_j$ is non-empty.

Case of a point:

Let $C = (M, D)$ be a parametric state-class of a *PITPN*, $x = [\lambda_1 \ldots \lambda_l\ \theta_1 \ldots \theta_n]^\top$ be a point of D and t_f be a transition firable from $(M, \{x\})$. We can formally define the successor of $\{x\}$ by the firing t_f from marking M by:

$$\mathsf{next}(\{x\}, t_f) = \left\{ \forall i \in [1..n] \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_l \\ \theta'_1 \\ \vdots \\ \theta'_n \end{bmatrix} \left| \begin{array}{l} \theta'_i \in J_s(t_i)(\nu) \text{ if } \uparrow\mathsf{enabled}\,(t_i, M, t_f) \\ \theta'_i = \theta_i \text{ if } t_i \in \mathsf{enabled}\,(M) \\ \ \text{and } t_i \in \mathsf{inhibited}\,(M) \\ \ \text{and not } \uparrow\mathsf{enabled}\,(t_i, M, t_f) \\ \theta'_i = \theta_i - \theta_f \text{ otherwise} \end{array} \right. \right\}$$

The next operator straightforwardly extends to finite or infinite unions of points which allows to define formally $\mathsf{next}(D, t_f)$.

Parametric state-class graph:

The parametric state-class graph is generated by iteratively applying the function that computes the successors of a state-class:

Definition 9. Given a *PITPN* $\mathcal{N}$, the *parametric state-class graph* of $\mathcal{N}$ is the transition system $\mathcal{G}(\mathcal{N}) = \langle \mathcal{C}, \twoheadrightarrow, C_0 \rangle$ such that:

– $C_0 = (M_0, D_0)$ is the initial class such that $D_0 = D_p \wedge \{\theta_k \in J_s(t_k) \mid t_k \in \mathsf{enabled}\,(M_0)\}$

– $C \overset{t}{\twoheadrightarrow} C'$ iff $t \in \mathsf{firable}\,(C)$ and $C' = succ(C, t)$,

– $\mathcal{C} = \{C | C_0 \twoheadrightarrow^* C\}$, where $\twoheadrightarrow^*$ is the reflexive and transitive closure of $\twoheadrightarrow$.

Example 3. In the *PITPN* of the figure 1, we can exhibit the two following classes:

$$\begin{array}{lcl} \mathbf{C_0} = (\mathbf{M_0}, \mathbf{D_0}): & & \mathbf{C_1} = (\mathbf{M_1}, \mathbf{D_1}): \\ M_0 = (A, B) & & M_1 = (B, C) \\ D_0 = \begin{cases} 0 \leq a \leq \theta_1 \leq 10, \\ 0 \leq b \leq \theta_2 \leq c, \\ \theta_3 = 5. \end{cases} & \xrightarrow{t_1} & D_1 = \begin{cases} 0 \leq b \leq \theta_2 \leq c, \\ 0 \leq \theta_3 \leq 5 - a, \\ 0 \leq a. \end{cases} \end{array}$$

C_0 is the initial class in which transition t_1 is firable. After the firing of t_1 the class C_1 is reached. We now notice that to reach C_1 in the domain D_1 the parameter a must be lower than 5, which was not the case in the example 2 for the valuation $(6, 2, 3)$.

4.3 Valuation of the Parametric State-Class Graph

From the parametric state-class graph of a *PITPN* it is possible to choose a valuation of the parameters and to replace in the graph all the parameters by their value. Then, we obtain a non-parametric graph. However, some firing domains of the classes may become empty, which means that the class is not reachable for this valuation. Those classes must be removed from the non-parametric graph. The graph finally obtained corresponds to the state-class graph of the *ITPN* obtained for this valuation.

Definition 10 Valuation of a Parametric State-Class. Let $C = (M, D)$ be a parametric state-class of a *PITPN* $\mathcal{N}$ and let $\nu \in D_p$ be a valuation of the parameters of $\mathcal{N}$. The valuation of C by ν is a non-parametric class $[\![C]\!]_\nu = (M, [\![D]\!]_\nu)$ where:

$$[\![D]\!]_\nu = \{\nu' \in \mathbb{R}^n \mid (\nu|\nu') \in D\}$$

The valuation of the parametric state-class graph is obtained by valuating the classes of the graph, starting from the initial class and stopping if the firing domains become empty.

Definition 11 Valuation of the Parametric State-Class Graph. Given a *PITPN* $\mathcal{N}$ and a valuation $\nu \in D_p$, $[\![\mathcal{G}(\mathcal{N})]\!]_\nu = (\mathcal{C}_\nu, \twoheadrightarrow, [\![C_0]\!]_\nu)$ where:

- $[\![C_0]\!]_\nu$ is the valuation of the initial class C_0 of $\mathcal{G}(\mathcal{N})$,
- $[\![C]\!]_\nu \overset{t}{\twoheadrightarrow} [\![C']\!]_\nu$ iff $C = (M, D), C' = (M', D') \in \mathcal{G}(\mathcal{N})$ and $C \overset{t}{\twoheadrightarrow} C'$ and $[\![D']\!]_\nu \neq \emptyset$
- $\mathcal{C}_\nu = \{[\![C]\!]_\nu \mid [\![C_0]\!]_\nu \twoheadrightarrow^* [\![C]\!]_\nu\}$, where $\twoheadrightarrow^*$ is the reflexive and transitive closure of $\twoheadrightarrow$.

The theorem 12 establishes that the valuation of the parametric state-class graph of a *PITPN* matches the non-parametric state-class graph of the *ITPN* obtained for the same parameters valuation.

Theorem 12. *Given a PITPN $\mathcal{N}$ and a valuation $\nu \in D_p$, then*

$$[\![\mathcal{G}(\mathcal{N})]\!]_\nu = \mathcal{G}([\![\mathcal{N}]\!]_\nu)$$

where $\mathcal{G}([\![\mathcal{N}]\!]_\nu)$ is the non-parametric state-class graph of $[\![\mathcal{N}]\!]_\nu$.

Proof. By definition $[\![\mathcal{G}(\mathcal{N})]\!]_\nu = (\mathcal{C}_\nu, \twoheadrightarrow, [\![C_0]\!]_\nu)$.
The construction of $\mathcal{G}([\![\mathcal{N}]\!]_\nu)$ corresponds to the one of $[\![\mathcal{G}(\mathcal{N})]\!]_\nu$ when there is no parameters. We note $\mathcal{G}([\![\mathcal{N}]\!]_\nu) = (\mathcal{C}^*, \twoheadrightarrow, C_0^*)$.

1. First we show that $[\![C_0]\!]_\nu = C_0^*$:
   On the one hand, $[\![C_0]\!]_\nu = (M_0, [\![D_0]\!]_\nu)$ where $D_0 = D_p \wedge \{\theta_k \in J_s(t_k) \mid t_k \in \mathsf{enabled}\,(M_0)\}$ And since $\nu \in D_p$ we get that $[\![D_0]\!]_\nu = \{\theta_k \in J_s(t_k)(\nu) \mid t_k \in \mathsf{enabled}\,(M_0)\}$.
   On the other hand, by definition $C_0^* = (M_0, D_0^*)$ where
   $D_0^* = \{\theta_k \in J_s(t_k)(\nu) \mid t_k \in \mathsf{enabled}\,(M_0)\}$.

2. Let be $[\![C]\!]_\nu \in [\![\mathcal{G}(\mathcal{N})]\!]_\nu$ and $C^* \in \mathcal{G}([\![\mathcal{N}]\!]_\nu)$. If we assume that $[\![C]\!]_\nu = C^*$ (we now confound M and M^* and only use M), we can induce that $\forall t \in \mathsf{enabled}\,(M)$, $(\exists C'$ s.t. $[\![C]\!]_\nu \overset{t}{\twoheadrightarrow} [\![C']\!]_\nu \iff \exists C^{**}$ s.t. $C^* \overset{t}{\twoheadrightarrow} C^{**})$ and $[\![C']\!]_\nu = C^{**}$:

   (a) If $[\![C]\!]_\nu \overset{t}{\twoheadrightarrow}_\nu [\![C']\!]_\nu$ then $\begin{cases} C = (M, D), C' = (M', D') \in \mathcal{G}(\mathcal{N}) \\ C \overset{t}{\twoheadrightarrow} C' \\ [\![D']\!]_\nu \neq \emptyset \end{cases}$

   Since $[\![D']\!]_\nu \neq \emptyset$ and $C \overset{t}{\twoheadrightarrow} C'$, it means that t is firable in C for ν, i.e. $\exists \nu'$, s.t. $(\nu|\nu') \in D$ and $\forall j \in \{1, \dots, n\} - \{i\}$, s.t. t_j is active, $\nu'(\theta_i) \leq \nu'(\theta_j)$. Thus, we deduce that $\nu' \in [\![D]\!]_\nu = D^*$, and naturally that $t \in \mathsf{firable}\,(C^*)$. So there exists C^{**} such that $C^* \overset{t}{\twoheadrightarrow} C^{**}$.

   (b) Conversely, if $C^* \overset{t}{\twoheadrightarrow} C^{**}$ then $t \in \mathsf{firable}\,(C^*)$ and so $\exists \nu' \in D^*$ such that $\forall j \in \{1, \dots, n\} - \{i\}$, s.t. t_j is active, $\nu'(\theta_i) \leq \nu'(\theta_j)$. Since $\nu' \in D^* = [\![D]\!]_\nu$ it follows that $(\nu|\nu') \in D$ and consequently we immediately deduce that $t \in \mathsf{firable}\,(C)$. Thus, there exists C' such that $C \overset{t}{\twoheadrightarrow} C'$ and since $t \in \mathsf{firable}\,(C)$ for the point $(\nu|\nu')$ it means that $[\![D']\!]_\nu \neq \emptyset$.

   Finally, we have to prove that $C^{**} = [\![C']\!]_\nu$. Immediately, we get that $M^{**} = M'$. Concerning the two firing domains, we have $D' = \bigcup_{x \in D} \mathsf{next}(\{x\}, t)$. On the one hand, $[\![D']\!]_\nu = \bigcup_{x = [\underbrace{\lambda_1 \dots \lambda_l}_{\nu}\ \theta_1 \dots \theta_n] \in D} \mathsf{next}(\{x\}, t)$. On the other hand, $D^{**} = \bigcup_{x = [\theta_1 \dots \theta_n] \in D^*} \mathsf{next}(\{x\}, t)$. And if $x = [\theta_1 \dots \theta_n] \in D^*$ since $D^* = [\![D]\!]_\nu$ it means that $[\lambda_1 \dots \lambda_l\ \theta_1 \dots \theta_n] \in D$ (and reciprocally). Consequently, since the next operator used is the same in the parametric and non-parametric case (it does not modify the parameters) the two firing domains are equals. □

Finally, the theorem 13 and the lemma 14 allow to directly compute the reachability of a class for a valuation of the parameters by checking the accessibility condition of the parametric state-class.

Theorem 13. *Given a PITPN $\mathcal{N}$ and a valuation $\nu \in D_p$, let $C = (M, D)$ be a parametric state-class in $\mathcal{G}(\mathcal{N})$. Then*

$$[\![C]\!]_\nu \in [\![\mathcal{G}(\mathcal{N})]\!]_\nu \textit{ iff } \nu \in D_{|Par}$$

$D_{|Par}$ is called the accessibility condition *of C.*

This theorem relies on the fact that the accessibility condition of a mother class always entails the accessibility conditions of its successors, which is expressed by the lemma 14.

Lemma 14. *Given a PITPN $\mathcal{N}$, let $C = (M, D)$ and $C' = (M', D')$ be two parametric state-classes in $\mathcal{G}(\mathcal{N})$. If $C \overset{t}{\twoheadrightarrow} C'$ then $D'_{|Par} \subseteq D_{|Par}$.*

Proof of Lemma 14. Let be $\nu \in D'_{|Par}$. Then by definition of the projection $\exists \nu' \in \mathbb{R}^{n'}$ (where n' is the number of enabled transitions in C') such that $(\nu|\nu') \in D'$. Because $C' = succ(C, t)$, this implies that $D' = \mathsf{next}(D, t)$. And now $(\nu|\nu') \in D'$ implies that $\exists \nu'' \in \mathbb{R}^n$ (where n is the number of enabled transitions in C) such that $(\nu|\nu'') \in D$ and $\mathsf{next}(\{(\nu|\nu'')\}, t) = \{(\nu|\nu')\}$, because the parameters are not modified by the next operator. Again, by definition of the projection, this implies that $\nu \in D_{|Par}$ which proves the lemma. $\square$

Proof of Theorem 13. We directly deduce that if $[\![C]\!]_\nu \in [\![\mathcal{G}(\mathcal{N})]\!]_\nu$ then $[\![D]\!]_\nu \neq \emptyset \Rightarrow \exists \nu' \in \mathbb{R}^n$ s.t. $(\nu|\nu') \in D \Rightarrow \nu' \in D'_{|Par}$.
Conversely, since $C \in \mathcal{G}(\mathcal{N})$ there exists $\{C_1, \ldots, C_k - 1\} \in \mathcal{G}(\mathcal{N})$ such that $C_0 \twoheadrightarrow C_1 \twoheadrightarrow \cdots \twoheadrightarrow C_{k-1} \twoheadrightarrow C_k = C$. Then, thanks to the lemma 14 we have the following inclusions: $D_{|Par} = D_{k|Par} \subseteq D_{k-1|Par} \subseteq \cdots \subseteq D_{1|Par} \subseteq D_{0|Par}$. If $\nu \in D_{|Par}$, then $\forall 0 \leq i \leq k,\ \nu \in D_{i|Par}$, which implies that $[\![D_i]\!]_\nu \neq \emptyset$. With these last conditions, starting from C_0, we can recursively prove that $[\![C_i]\!]_\nu \in [\![\mathcal{G}(\mathcal{N})]\!]_\nu$, especially for $i = k$ $[\![C]\!]_\nu \in [\![\mathcal{G}(\mathcal{N})]\!]_\nu$. $\square$

5 Parametric Model-Checking

The model-checking problem consists in checking that a model $\mathcal{N}$ satisfies a property ϕ expressed in a given logic, which is more formally written $\mathcal{N} \models \phi$. The answer to this problem is either *true* or *false*.

Given a parametric model $\mathcal{N}$ and a property ϕ, which may also be parameterized, we address the *parametric synthesis problem i.e.* we want to determine the set of parameters valuations $\Gamma(\mathcal{N}, \phi)$ such that $\forall \nu \in \Gamma(\mathcal{N}, \phi)$ the non-parametric model $[\![\mathcal{N}]\!]_\nu$ obtained for the valuation ν satisfies the non-parametric property $[\![\phi]\!]_\nu$ obtained for the same valuation, which is formally written $[\![\mathcal{N}]\!]_\nu \models [\![\phi]\!]_\nu$. This set will be represented by a set of constraints on the parameters of the problem.

5.1 Parametric TCTL Formulae

Like *ITPN*, we parameterize TCTL formulae by allowing that the bounds of the temporal intervals of the formulae are parameters. The parameters used in the formulae are added to the set of parameters of the *PITPN* in study. Besides, we consider only a subset of TCTL formulae for which "on-the-fly" model-checking algorithms have already been proposed for *TPNs* [Hadjidj and Boucheneb, 2006]. This subset is sufficient to verify many interesting problems (reachability, safety, bounded liveness. . .).

First, we recall the syntax and semantics of TCTL formulae in the context of *TPNs* (or *ITPNs*).

Definition 15 TCTL for TPN. The grammar of TCTL formulae is:

$$TCTL ::= \mathcal{P} \mid \neg\varphi \mid \varphi \Rightarrow \psi \mid \exists\varphi\mathcal{U}_I\psi \mid \forall\varphi\mathcal{U}_I\psi$$

where $\varphi, \psi \in TCTL$, $I \in \mathcal{I}(\mathbb{Q}^+)$, $\mathcal{P} \in PR$, and $PR = \{\mathcal{P} \mid \mathcal{P} : M \to \{true, false\}\}$ is the set of propositions on the markings of the net.

We use the following abbreviations $\exists\Diamond_I\varphi = \exists\mathbf{true}U_I\varphi$, $\forall\Diamond_I\varphi = \forall\mathbf{true}U_I\varphi$, $\exists\Box_I\varphi = \neg\forall\Diamond_I\neg\varphi$ and $\forall\Box_I\varphi = \neg\exists\Diamond_I\neg\varphi$.

We define the bounded time response by $\varphi \leadsto_I \psi = \forall\Box(\varphi \Rightarrow \forall\Diamond_I\psi)$.

TCTL formulae are interpreted on the states of a model $\mathcal{M} = (\mathcal{S}_\mathcal{N}, \mathcal{V})$, where $\mathcal{S}_\mathcal{N}$ is the state space of the *TPN* and $\mathcal{V} : \mathcal{S}_\mathcal{N} \to 2^{PR}$ is a function that evaluates the marking of a state, such that $\mathcal{V}(q) = \{\mathcal{P} \in PR \mid \mathcal{P}(M) = true\}$. Now, let $q \in \mathcal{S}_\mathcal{N}$ be a state and $\rho \in \pi(q)$ a run starting from q, such that $\rho = q^0 \xrightarrow{d_0} q^0 + d_0 \xrightarrow{t_0} q^1 \xrightarrow{d_1} q^1 + d_1 \xrightarrow{t_1} \cdots$. We define $\rho^* : \mathbb{R}^+ \to \mathcal{S}_\mathcal{N}$ by $\rho^*(r) = q^i + \delta$ if $r = \sum_{j=0}^{i-1} d_j + \delta$, with $i \geq 0$ and $0 \leq \delta < d_i$.

Definition 16 Semantics of TCTL. Given a *TPN* $\mathcal{N}$ and its model $\mathcal{M} = (\mathcal{S}_\mathcal{N}, \mathcal{V})$, the truth value of a TCTL formula for a state $q \in \mathcal{S}_\mathcal{N}$ is:

- $q \models \mathcal{P}$ iff $\mathcal{P} \in \mathcal{V}(q)$,
- $q \models \neg\varphi$ iff $q \not\models \varphi$,
- $q \models \varphi \Rightarrow \psi$ iff $q \not\models \varphi \ \vee q \models \psi$,
- $q \models \exists\varphi\mathcal{U}_I\psi$ iff $\exists\rho \in \pi(q),\ \exists r \in I$ s.t. $\rho^*(r) \models \psi$ and $\forall r' < r,\ \rho^*(r') \models \varphi$
- $q \models \forall\varphi\mathcal{U}_I\psi$ iff $\forall\rho \in \pi(q),\ \exists r \in I,$ s.t. $\rho^*(r) \models \psi$ and $\forall r' < r,\ \rho^*(r') \models \varphi$

Given a model $\mathcal{M} = (\mathcal{S}_\mathcal{N}, \mathcal{V})$, for a markings proposition $\mathcal{P} \in PR$ and a state $q = (M, I) \in \mathcal{S}_\mathcal{N}$, we use the notation $M \models \mathcal{P}$ if $\mathcal{P} \in \mathcal{V}(q)$ and $M \not\models \mathcal{P}$ if $\mathcal{P} \notin \mathcal{V}(q)$.

Finally, a *TPN* $\mathcal{N}$ satisfies a TCTL formula ϕ if and only if $q_0 \models \phi$.

We present now the syntax and semantics of Parametric TCTL (PTCTL) formulae for *PITPN*.

Definition 17 PTCTL for PITPN. The grammar of PTCTL formulae is:

$$PTCTL ::= \exists\varphi\mathcal{U}_J\psi \mid \forall\varphi\mathcal{U}_J\psi \mid \exists\Diamond_J\varphi \mid \forall\Diamond_J\varphi \mid \exists\Box_J\varphi \mid \forall\Box_J\varphi \mid \varphi \leadsto_{J_r} \psi$$

where $\varphi, \psi \in PR$, $J, J_r \in \mathcal{J}(Par)$ are parametric time intervals, with the restriction that $J_r = [0, b]$ with $b \in \mathbb{Q}^+ \cup Par$, or $J_r = [0, \infty[$.

The semantics of PTCTL formulae are defined similarly to the semantics of *PITPNs*. Given a valuation, the parameters in the formulae are replaced by their value to obtain a TCTL formula, which is interpreted on the *ITPN* obtained for this valuation.

Definition 18 Semantics of PTCTL. Let $\mathcal{N}$ be a *PITPN* and ϕ be a PTCTL formulae and $\nu \in D_p$ be a valuation of the parameters of $\mathcal{N}$ (which are shared with ϕ). $[\![\phi]\!]_\nu$ is the TCTL formula obtained when replacing in ϕ the parametric time interval J (or J_r) by the $\mathbb{Q}$-interval $J(\nu)$ (or $J_r(\nu)$).
Then $\mathcal{N}$ satisfy ϕ for the valuation ν if and only if $[\![\mathcal{N}]\!]_\nu \models [\![\phi]\!]_\nu$.

5.2 Extending the Parametric State-Class Graph with a Global Clock

In the state-class graph, the firing domain of a class gives the firing dates of the transitions with the entrance in the class as a time origin. Timed properties are difficult to verify in this context. In order to easily check timed properties with the state-class graph abstraction, it is necessary to be able to evaluate the time that has elapsed between classes. For this purpose, we propose to extend the parametric state-classes with an additional variable noted θ_c. This variable is initialized to zero in the initial class, and then decreases when time flows, like a transition variable [1] . However, the variable will not constrain the transitions variables when determining the firability constraints. Then, for all classes, the time elapsed from the initialization of θ_c to the entrance in the class is: $\tau_c = -\theta_c$.

Definition 19. An *extended parametric state-class* C of a *PITPN* is a class whose firing domain D is extended with an additional variable $\theta_c \in \Theta$.

The definition of the firability of an extended class is not modified. The firability constraints indeed only involve the variables θ_i where $\forall i \in \{1, \dots, n\}$, $t_i \in T$. The next operator is redefined for an extended class such that for a point $x = [\lambda_1 \dots \lambda_l\ \theta_1 \dots \theta_n\ \theta_c]^\top$ of D, in $\mathsf{next}((\{x\}, t_f)$ we have $\theta'_c = \theta_c - \theta_f$.

[1] The value of this variable will always be non-positive. But this is not a problem in the computation of the state-classes. The alternative would be to initialize it, not to zero, but to a sufficiently large value, but this value is hard to determine.

The extended parametric state-class graph $\mathcal{G}_c(\mathcal{N})$ is then computed iteratively in a similar way, starting from the initial class $C_0 = (M_0, D_0)$ where

$$D_0 = D_p \wedge \{\theta_k \in J(t_k) \mid t_k \in \mathsf{enabled}\,(M_0)\} \wedge \{\theta_c = 0\}$$

Finally, given an extended parametric state-class $C = (M, D)$, we are able to determine:

- $\tau_{min}(C)$, the absolute minimum time elapsed when entering the class. This is a function of the parameters Par of the net $\tau_{min}(C) : D_p \to \mathbb{Q}^+$, such that $\tau_{min}(C)(\nu) = \min_{x=(\nu|\nu')\in D}(\tau_c)$. It can be expressed as the maximum between the minimum values of τ_c and it is necessarily positive and finite.

- $\tau_{max}(C)$, the absolute maximum time elapsed when entering the class. This is a function on the parameters Par of the net $\tau_{max}(C) : D_p \to \mathbb{Q}^+ \cup \{\infty\}$, such that $\tau_{max}(C)(\nu) = \max_{x=(\nu|\nu')\in D}(\tau_c)$. It can be expressed as the minimum between the maximum values of τ_c and it is necessarily positive but may be infinite if there is no maximum time.

If $[\![C]\!]_\nu \in [\![\mathcal{G}_c(\mathcal{N})]\!]_\nu$, let $q \in [\![C]\!]_\nu$ be a state. Then the elapsed time of the state is such that $\tau_{min}(C)(\nu) \leq \mathsf{time}(q) \leq \tau_{max}(C)(\nu)$.

Example 4. If we consider the two classes previously presented in the example 3 for the *PITPN* of the figure 1, the corresponding extended classes are:

$$\begin{array}{l} \mathbf{C_0} = (\mathbf{M_0}, \mathbf{D_0}) : \\ M_0 = (A, B) \\ D_0 = \begin{cases} 0 \leq a \leq \theta_1 \leq 10, \\ 0 \leq b \leq \theta_2 \leq c, \\ \theta_3 = 5, \theta_c = 0. \end{cases} \end{array} \xrightarrow{t_1} \begin{array}{l} \mathbf{C_1} = (\mathbf{M_1}, \mathbf{D_1}) : \\ M_1 = (B, C) \\ D_1 = \begin{cases} \theta_3 - \theta_c = 5, \\ 0 \leq b \leq \theta_2 \leq c, \\ -5 \leq \theta_c \leq -a, \\ 0 \leq a. \end{cases} \end{array}$$

Thus, the elapsed time after the firing of t_1 is such that $\tau_{min}(C_1) = a$ and $\tau_{max}(C_1) = 5$. We notice that since t_2 was inhibited in class C_0 its clock value has not changed unlike the one of t_3.

5.3 Principles of Parametric Model-Checking with the State-Class Graph

Given a *PITPN* $\mathcal{N}$ and a PTCTL property ϕ, we want to characterize the set $\Gamma(\mathcal{N}, \phi)$ of all the parameters valuations that solve the problem, which is defined by:

$$\Gamma(\mathcal{N}, \phi) = \{\nu \in D_p \mid [\![\mathcal{N}]\!]_\nu \models [\![\phi]\!]_\nu\}$$

To achieve this we are going to recursively compute, on each extended class $C = (M, D)$, a logical predicate on the parameters that corresponds to the verification of the property on the current class and its successors. This predicate represents the set: $F^{\phi}(C) = \{\nu \in D_{|Par} \mid [\![C]\!]_\nu \models [\![\phi]\!]_\nu\}$.

We begin by giving an interpretation of the verification of a PTCTL formula ϕ on an extended parametric state class C, which we write $[\![C]\!]_\nu \models [\![\phi]\!]_\nu$.

For formulae $\phi = \exists\varphi\mathcal{U}_J\psi$ or $\phi = \forall\varphi\mathcal{U}_J\psi$:

For a valuation $\nu \in D_{|Par}$ and a state $q \in [\![C]\!]_\nu$, we define $[\![\phi[J - time(q)]]\!]_\nu$ as the TCTL formula obtained after replacing in ϕ the parametric time interval J by $J(\nu) - time(q)$.

Then, according to the form of the PTCTL formula ϕ we define that:

- if $\phi = \exists\varphi\mathcal{U}_J\psi$, then $[\![C]\!]_\nu \models [\![\phi]\!]_\nu$ iff $\exists q \in [\![C]\!]_\nu,\ q \models [\![\phi[J - \mathsf{time}(q)]]\!]_\nu$
- if $\phi = \forall\varphi\mathcal{U}_J\psi$, then $[\![C]\!]_\nu \models [\![\phi]\!]_\nu$ iff $\forall q \in [\![C]\!]_\nu,\ q \models [\![\phi[J - \mathsf{time}(q)]]\!]_\nu$

For formulae $\phi = \varphi \leadsto_{J_r} \psi$:

We extend the *PITPN* $\mathcal{N}$ with an additional place named P_{LT} that will be marked iff we are looking for ψ. We denote by $\mathcal{N}_{LT}$ the resulting *PITPN*. In this model, the successor $C' = (M', D') = succ_{LT}(C, t)$ of an extended parametric state-class $C = (M, D) \in \mathcal{G}_c(\mathcal{N}_{LT})$ by a transition $t_f \in \mathsf{firable}(C)$, is given by:

- $M' = M - {}^\bullet t_f + t_f^\bullet$ and $\begin{cases} \text{if } (M' \models \varphi \text{ and } M' \not\models \psi) \text{ then } M'(P_{LT}) = 1, \\ \text{else if } (M \models \psi) \text{ then } M'(P_{LT}) = 0, \\ \text{else } M'(P_{LT}) = M(P_{LT}) \end{cases}$
- $D' = \mathsf{next}(D, t_f)$ and
  if $(M(P_{LT}) = 0$ or $M \models \psi)$ then the clock variable θ_c is reset to zero.

On this model we define that $[\![C]\!]_\nu \models [\![\phi]\!]_\nu$ iff $\forall q \in [\![C]\!]_\nu,\ \forall \rho \in \pi(q)$,

$$M(P_{LT}) = 1 \Rightarrow \begin{cases} \exists 0 \leq r_1 \leq J_r(\nu)^{\downarrow} - \mathsf{time}(q) \text{ s.t. } \rho^*(r_1) \models \psi \text{ and} \\ \forall r_2 \geq r_1\ \rho^*(r_2) \models M(P_{LT}) = 1 \Rightarrow \exists r_3 \geq r_2 \\ \qquad\qquad \text{s.t. } r_3 - r_2 \leq J_r(\nu)^{\downarrow} \text{ and } \rho^*(r_3) \models \psi \end{cases}$$
$$M(P_{LT}) = 0 \Rightarrow \begin{cases} \forall r_2 \geq 0\ \rho^*(r_2) \models M(P_{LT}) = 1 \Rightarrow \exists r_3 \geq r_2 \\ \qquad \text{s.t. } r_3 - r_2 \leq J_r(\nu)^{\downarrow} \text{ and } \rho^*(r_3) \models \psi \end{cases}$$

In this model, $\mathsf{time}(q)$ refers to the time elapsed since the last reinitialization of θ_c. We notice that when the time has been reset (then $\mathsf{time}(q) = 0$) the two definitions above are equivalent and correspond to $q \models [\![\phi]\!]_\nu$.

Finally, the theorem 20 states that we are able to resolve the parametric model-checking problem if we compute the set of solutions on the initial class.

Theorem 20. *Given a PITPN $\mathcal{N}$ and a PTCTL formula ϕ, $\Gamma(\mathcal{N},\phi) = F^{\phi}(C_0)$, where C_0 is the initial class of the extended parametric state class graph of $\mathcal{N}$.*

Proof. According to their respective definitions we have:

On the one hand $\Gamma(\mathcal{N},\phi) = \{\nu \in D_p \mid [\![\mathcal{N}]\!]_\nu \models [\![\phi]\!]_\nu\}$ and by definition: $[\![\mathcal{N}]\!]_\nu \models [\![\phi]\!]_\nu \Leftrightarrow q_0 \models [\![\phi]\!]_\nu$, where q_0 is the initial state of $[\![\mathcal{N}]\!]_\nu$.

On the other hand, for $C_0 = (M_0, D_0)$:

– if $\phi = \exists\varphi\mathcal{U}_J\psi$ or $\phi = \forall\varphi\mathcal{U}_J\psi$, then $F^{\phi}(C_0) = \{\nu \in D_{0|Par} \mid q_0 \models [\![\phi[I - \mathsf{time}(q_0)]]\!]_\nu\}$, because the initial class $[\![C_0]\!]_\nu$ matches the initial state $q_0 = (M_0, J_s(\nu))$. Moreover, $D_{0|Par} = D_p$ and $\mathsf{time}(q_0) = 0$.

– if $\phi = \varphi \leadsto_{J_r} \psi$ then similarly $[\![C_0]\!]_\nu$ matches the initial state q_0 and $\mathsf{time}(q_0) = 0$. Consequently, the previous definition of $F^{\phi}(C_0)$ corresponds to the definition of $q_0 \models [\![\phi]\!]_\nu$.

Thus, whatever the form of ϕ the two sets are syntactically equal. □

5.4 Parametric Model-Checking Semi-Algorithms

To verify PTCTL formulae we propose three semi-algorithms according to the form of the formulae. These algorithms recursively characterize for each class C the set $F^{\phi}(C)$. This set is represented by conjunctions or disjunctions of linear constraints on the parameters. We use a disjunctive normal form (i.e. a disjunction of convex polyhedra).

The proofs of the correctness and the completeness of these semi-algorithms can be found in the appendix A.

5.4.1 Algorithm EU:

This semi-algorithm is designed for formulae whose form is $\phi = \exists\varphi\mathcal{U}_J\psi$, where $J \in \mathcal{J}(Par)$. Let be $C = (M, D) \in \mathcal{G}_c(\mathcal{N})$, we compute:

$$\begin{aligned}
F^{\phi}_{EU}(C) = & D_{|Par} \wedge \{\tau_{min}(C) \leq J^{\downarrow}\} \wedge \Bigg(\Big(M \models \psi \wedge \{\tau_{max}(C) \geq {}^{\uparrow}J\}\Big) \\
& \vee \Bigg(M \models \varphi \wedge M \models \psi \wedge \Big(\mathsf{firable}(C) = \emptyset \vee \\
& \qquad \Big(\bigvee_{\substack{t \in \mathsf{firable}(C)\\ C'=(M',D')=succ(C,t)}} (\{\tau_{max}(C') \geq {}^{\uparrow}J\} \wedge D'_{|Par})\Big)\Big)\Bigg) \\
& \vee \Bigg(M \models \varphi \wedge \mathsf{firable}(C) \neq \emptyset \wedge \Big(\bigvee_{\substack{t \in \mathsf{firable}(C)\\ C'=succ(C,t)}} F^{\phi}_{EU}(C')\Big)\Bigg)\Bigg)
\end{aligned}$$

The first two conditions $D_{|Par} \ \wedge \ \{\tau_{min}(C) \leq J^{\downarrow}\}$ ensure that the class is accessible and that the maximum time of the formula ϕ is not overpassed. Then, three conditions in disjunction allow to prove the formula ϕ:

– The first disjunction is used when C verifies ψ but not ϕ. Thus, the elapsed time must be entailed in the interval J as soon as it gets into the class.

– The second case is used when both ϕ and ψ are verified. Comparing to the first one, it is less restrictive since it allows to wait in the class.

– The third disjunction is used whenever ϕ is verified. In this case the successors of C are computed.

Example 5. In the net of the figure 1 we check the formula:
$\phi_1 = \exists\Diamond_{[0,inf[}(M(D) = 1)$. The result is $F^{\phi_1}_{EU}(C_0) = \{a + b <= 5\}$.

5.4.2 Algorithm AU:

This semi-algorithm is designed for formulae whose form is $\phi = \forall\varphi\mathcal{U}_J\psi$, where $J \in \mathcal{J}(Par)$. Let be $C = (M, D) \in \mathcal{G}_c(\mathcal{N})$, we compute:

$$
\begin{aligned}
F^{\phi}_{AU}(C) = D_{|Par} \ \wedge \ &\left\{\begin{matrix} \tau_{max}(C) \leq J^{\downarrow} \\ \tau_{max}(C) \neq \infty \end{matrix}\right\} \wedge \ \left(\left(M \models \psi \ \ \wedge \{\tau_{min}(C) \geq {}^{\uparrow}J\}\right)\right. \\
\vee \Bigg(M \models \varphi \ \wedge \ M \models \psi \ \wedge \ \Bigg(&\mathsf{firable}\,(C) = \emptyset \ \vee \\
&\Big(\bigwedge_{\substack{t \in \mathsf{firable}(C) \\ C'=(M',D')=succ(C,t) \\ D''=D' \wedge \{\theta_c > -{}^{\uparrow}J\}}} (F^{\phi}_{AU}(M', D'') \vee \neg D''_{|Par})\Big)\Bigg)\Bigg) \\
\vee \Bigg(M \models \varphi \ \wedge \ \mathsf{firable}\,(C) \neq \emptyset \ \wedge \ \Big(& \bigwedge_{\substack{t \in \mathsf{firable}(C) \\ C'=(M',D')=succ(C,t)}} \big(F^{\phi}_{AU}(C') \vee \neg D'_{|Par}\big)\Big)\Bigg)\Bigg)
\end{aligned}
$$

It uses similar but stricter conditions than the previous algorithm. For instance, after the accessibility condition, to check that the maximum time of ϕ is not overpassed the maximum elapsed time is now used (and respectively, to check if the minimum time of ϕ is reached, the minimum elapsed time is used). Then, a disjunction between three conditions must be verified. Unlike previously, in the second one some successors must computed, but only for the points of the class for which the property has not been verified yet. When iterating the algorithm on successors, either the formula is verified on the successor, or the condition $\neg D'_{|Par}$ (or $\neg D''_{|Par}$) forbid the accessibility of this successor.

Example 6. In the net of the figure 1 we check the formula:
$\phi_2 = \forall\Diamond_{[0,inf]}(M(E) = 1)$. The result is $F^{\phi_2}_{AU}(C_0) = \{a + b > 5\}$.

5.4.3 Algorithm LT:

This semi-algorithm is designed for formulae whose form is $\phi = \varphi \leadsto_{J_r} \psi$, where $J_r \in \mathcal{J}(Par)$ such that $J_r = [0, b]$ with $b \in \mathbb{Q}^+ \cup Par$, or $J_r = [0, \infty[$. Let be $C = (M, D) \in \mathcal{G}_c(\mathcal{N}_{LT})$, we compute:

$$F^{\phi}_{LT}(C) = D_{|Par} \ \wedge \ \left(M(P_{LT}) = 0 \ \vee \ \left\{ \begin{array}{l} \tau_{max}(C) \leq J_r^{\downarrow} \\ \tau_{max}(C) \neq \infty \end{array} \right\} \right)$$
$$\wedge \left(\left(\mathsf{firable}\,(C) = \emptyset \wedge \big(M(P_{LT}) = 0 \vee M \models \psi\big) \right) \vee \right.$$
$$\left. \left(\mathsf{firable}\,(C) \neq \emptyset \ \wedge (\bigwedge_{\substack{t \in \mathsf{firable}(C) \\ C'=(M',D')=succ_{LT}(C,t)}} (F^{\phi}_{LT}(C') \vee \neg D'_{|Par})) \right) \right)$$

This algorithm is similar to F_{AU} when $^{\uparrow}J = 0$. However, the analysis can only stop if no successor is found.

6 Discussion and implementation

The semi-algorithms presented in this paper have been implemented in Romeo [Lime et al., 2009], a software tool for time Petri nets analysis. Romeo provides a graphical user interface for editing and simulating time Petri nets and Petri nets with stopwatches (with or without parameters), and a computation module that performs TCTL model-checking and state-space computation. Although we have focused our presentation on *ITPNs*, Romeo allows the modelling of stopwatches either with inhibitor arcs or with priorities and the algorithms presented in this paper can be applied to both models.

For time Petri nets, efficient model-checking is performed, using the Uppaal DBM Library [Larsen et al., 1997]. For Petri nets with stopwatches and parametric time Petri nets, polyhedra manipulation are necessary. The Parma Polyhedra Library [Bagnara et al., 2006] is used to represent the firing domains of the parametric state-classes and the logical formulae computed by the parametric model-checking algorithms. These formulae are represented as powersets of convex polyhedra, that is to say a finite disjunction of polyhedra.

As mentioned before, the parametric model-checking problem is undecidable. Indeed the parametric state-class graph of a *PITPN* may be infinite. Additionally, to determine the whole set of parameters valuations that satisfy a formula, it would be in general necessary to analyze every parametric state-class. Nevertheless, some methods can help with the termination (and are necessary to the termination). In this way, if a parametric state-class $C = (M, D)$ is included in another class $C' = (M', D')$ (i.e. $M = M'$ and $D \subseteq D'$), it can be shown that $F^{\phi}_{EU}(C) \subseteq F^{\phi}_{EU}(C')$, and on the contrary that $F^{\phi}_{AU}(C') \subseteq F^{\phi}_{AU}(C) \vee \neg D_{|Par}$ and $F^{\phi}_{LT}(C') \subseteq F^{\phi}_{LT}(C) \vee \neg D_{|Par}$. As a result, in our “on-the-fly” model-checking approach it will not be necessary to analyze the whole state-class graph, but we will be able to stop the analysis of successors when finding included parametric state-classes.

7 Case study

We illustrate our method in a scheduling problem taken from [Bucci et al., 2004], that we parameterized by replacing some temporal constraints by parameters. We consider a system of three tasks: $task_1$ and $task_3$ are periodic, $task_2$ is sporadic. The periods are expressed in function of a time parameter a and are respectively a, $2.a$ and $3.a$ for the tasks 1, 2 and 3. The system has fixed priorities between the tasks: $task_1$ has the greatest priority, then $task_2$ and then $task_3$.

We design a *PITPN* model of this system in the tool Romeo and obtain the model presented in Figure 2. The inhibitors arcs, drawn with a circle end, are used to modelize the priorities between the tasks. Besides, we can restrict the domain of the parameter, so that $D_p = \{30 \leq a \leq 70\}$.

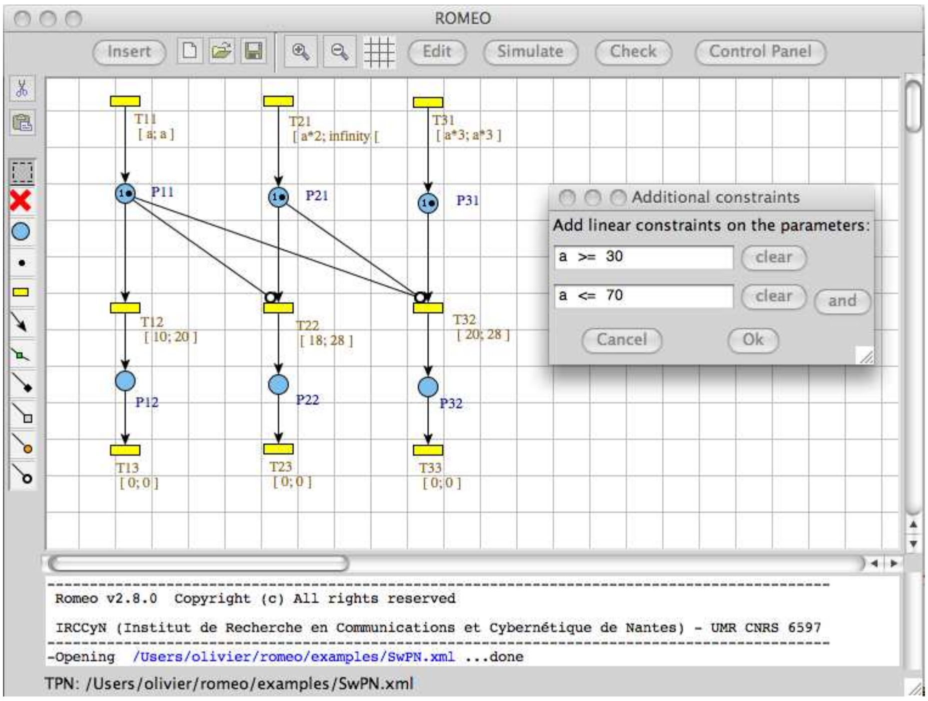

Figure 2: *PITPN* model in Romeo

The interesting model-checking problems on this system first concern the schedulability of the three tasks, which is expressed by the property that the *PITPN* model is safe (i.e. 1−bounded). The semi-algorithms presented in this paper and implemented in ROMEO allow to verify this property by checking the TCTL formula:

$$\forall P_i,\ \forall\Box_{[0,\infty[}(M(P_i) \leq 1)$$

The result of the parametric model-checking is $a > 48$.

This new constraint can be added to the domain D_p of the parameters, which assures that the parametric system is now schedulable for every values of a. New properties can then be checked. For instance, we can compute the worst case response time (WCRT) of $task_3$ with the parametric TCTL formulae:

$$M(P_{31}) > 0 \leadsto_{[0,b]} M(P_{32}) > 0$$

This formula uses a new parameter b that is a maximum bound for the WCRT. The result of the parametric model-checking for this formula is $b \geq 96$ and thus 96 is the WCRT of $task_3$. This is in accordance with [Bucci et al., 2004] in which $a = 50$. We remark that this result is proven on the parametric model and so applies for all the values of a in the domain D_p, although it does not depend on it.

8 Conclusion

In this paper, we have introduced a parametric extension of time Petri nets with stopwatches where the temporal bounds of the firing intervals are replaced by temporal parameters. We have proposed a symbolic representation of the state-space of these parametric models which is based on a parametric extension of the state-class graph. Upon this abstraction we have developed semi-algorithms for the parametric model-checking of parametric TCTL formulae.

In our future works we want to integrate this parametric approach in the development cycle of real-time systems through the functional decomposition of the systems. On concrete examples, a parametric decomposition combined with a projection of the formulae to verify can be useful in the development process. We hope to succeed in the elaboration of a formal framework for this method so that the process could be automated.

References

[Alur et al., 1993] Alur, R., Henzinger, T. A., and Vardi, M. Y. (1993). Parametric real-time reasoning. In *ACM Symposium on Theory of Computing*, pages 592–601.

[Bagnara et al., 2006] Bagnara, R., Hill, P. M., and Zaffanella, E. (2006). The Parma Polyhedra Library. Quaderno 457, Dipartimento di Matematica, Università di Parma, Italy.

[Bérard et al., 2005] Bérard, B., Cassez, F., Haddad, S., Lime, D., and Roux, O. H. (2005). Comparison of the expressiveness of timed automata and time Petri nets. In *FORMATS 05*, volume 3829 of *LNCS*, Uppsala, Sweden. Springer.

[Berthomieu and Diaz, 1991] Berthomieu, B. and Diaz, M. (1991). Modeling and verification of time dependent systems using time Petri nets. *IEEE trans. on Soft. Eng.*, 17(3):259–273.

[Berthomieu et al., 2007] Berthomieu, B., Lime, D., Roux, O. H., and Vernadat, F. (2007). Reachability problems and abstract state spaces for time petri nets with stopwatches. *Discrete Event Dynamic Systems*, 17(2):133–158.

[Bowden, 1996] Bowden, F. D. J. (Jul, 1996). Modelling time in petri nets. In *Proceedings of the second Australia-Japan Workshop on Stochastic Models in Engineering, Technology and Management, Gold Coast, Australia*.

[Bruyère and Raskin, 2007] Bruyère, V. and Raskin, J.-F. (2007). Real-time model-checking: Parameters everywhere. *CoRR*, abs/cs/0701138.

[Bucci et al., 2004] Bucci, G., Fedeli, A., Sassoli, L., and Vicario, E. (2004). Time state space analysis of real-time preemptive systems. *IEEE trans. on Soft. Eng.*, 30(2):97–111.

[Cassez et al., 2006] Cassez, F. and Roux, O. H. (2006). Structural translation from Time Petri Nets to Timed Automata – Model-Checking Time Petri Nets via Timed Automata. *The journal of Systems and Software*, 79(10):1456–1468.

[Hadjidj and Boucheneb, 2006] Hadjidj, R. and Boucheneb, H. (2006). On-the-fly tctl model checking for time petri nets using state class graphs. In *ACSD*, pages 111–122. IEEE Computer Society.

[Henzinger et al., 1997] Henzinger, T. A., Ho, P.-H., and Wong-Toi, H. (1997). HYTECH: A model checker for hybrid systems. *Int. Journal on Soft. Tools for Technology Transfer*, 1(1–2):110–122.

[Henzinger et al., 1994] Henzinger, T. A., Nicollin, X., Sifakis, J., and Yovine, S. (1994). Symbolic model checking for real-time systems. *Information and Computation*, 111:394–406.

[Hune et al., 2001] Hune, T., Romijn, J., Stoelinga, M., and Vaandrager, F. W. (2001). Linear parametric model checking of timed automata. In *TACAS*, volume 2031 of *LNCS*. Springer.

[Larsen et al., 1995] Larsen, K. G., Pettersson, P., and Yi, W. (1995). Model-checking for real-time systems. In *Fundamentals of Computation Theory*, pages 62–88.

[Larsen et al., 1997] Larsen, K. G., Pettersson, P., and Yi, W. (1997). Uppaal in a nutshell. *International Journal on Software Tools for Technology Transfer*, 1(1–2):134–152.

[Lime et al., 2009] Lime, D., Roux, O. H., Seidner, C., and Traonouez, L.-M. (2009). Romeo: A parametric model-checker for petri nets with stopwatches. In Kowalewski, S. and Philippou, A., editors, *15th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2009)*, volume 5505 of *Lecture Notes in Computer Science*, pages 54–57, York, United Kingdom. Springer.

[Merlin, 1974] Merlin, P. (1974). *A study of the recoverability of computing systems*. PhD thesis, Department of Information and Computer Science, Univ. of California, Irvine.

[Roux and Déplanche, 2002] Roux, O. and Déplanche, A.-M. (2002). A t-time Petri net extension for real time-task scheduling modeling. *European Journal of Automation (JESA)*, 36(7):973–987.

[Roux and Lime, 2004] Roux, O. H. and Lime, D. (2004). Time Petri nets with inhibitor hyperarcs. Formal semantics and state space computation. In *ICATPN'04*, volume 3099 of *LNCS*, pages 371–390, Bologna, Italy. Springer.

[Traonouez et al., 2008] Traonouez, L.-M., Lime, D., and Roux, O. H. (2008). Parametric model-checking of time petri nets with stopwatches using the state-class graph. In Cassez, F. and Jard, C., editors, *6th International Conference on Formal Modelling* *and Analysis of Timed Systems (FORMATS 2008)*, volume 5215 of *Lecture Notes in Computer Science*, pages 280–294, Saint-Malo, France. Springer.

[Virbitskaite and Pokozy, 1999] Virbitskaite, I. and Pokozy, E. (1999). Parametric behaviour analysis for time petri nets. In *PaCT '99*, pages 134–140, London, UK. Springer-Verlag.

[Wang, 1996] Wang, F. (1996). Parametric timing analysis for real-time systems. *Inf. Comput.*, 130(2):131–150.

A Appendix

We present in this appendix the proofs of the semi-algorithms introduced in the paper. For two states q^0, q^1, we use the following abbreviations $\mathsf{time}(q^0) = \tau_0$ and $\mathsf{time}(q^1) = \tau_1$.

A.1 Algorithm EU

Theorem 21. *Let $\mathcal{N}$ be a PITPN and $\phi = \exists\varphi\mathcal{U}_J\psi$ a PTCTL formula, where $J \in \mathcal{J}(Par)$. Given an extended parametric state-class $C = (M, D)$ of $\mathcal{G}_c(\mathcal{N})$, the correctness and completeness of the semi-algorithm EU are expressed by:*

$$F^{\phi}(C) = F^{\phi}_{EU}(C)$$

Proof of Theorem 21 (correctness). We consider a valuation that belongs to the computed logical formula: $\nu \in F^{\phi}_{EU}(C)$. We prove that this valuation is correct, which means that: $\nu \in F^{\phi}(C) = \{\nu \in D_{|Par} \mid \exists q^0 \in [\![C]\!]_\nu,\ q^0 \models [\![\phi[I - \tau_0]]\!]_\nu\}$.

We directly deduce that $\nu \in D_{|Par}$, and it implies that $[\![C]\!]_\nu \neq \emptyset$. We also note that $\nu \in \{\tau_{min}(C) \leq J^{\downarrow}\}$. Then, there exists 3 ways to verify the disjunctive formula:

1. if ν verifies the first member of the disjunction (i.e. $M \models \psi\ \wedge\ \{\tau_{max}(C) \geq J^{\uparrow}\}$).
   We choose q^0 s.t. $J(\nu)^{\uparrow} \leq \tau_0 \leq J(\nu)^{\downarrow}$, which is possible since $\nu \in (\{\tau_{min}(C) \leq J^{\downarrow}\} \wedge \{\tau_{max}(C) \geq J^{\uparrow}\})$, thus $[\tau_{min}(C)(\nu), \tau_{max}(C)(\nu)] \cap [J(\nu)^{\uparrow}, J(\nu)^{\downarrow}] \neq \emptyset$. Then, let be $\rho \in \pi(q^0)$, for $r = 0$, we have $\rho^*(r) \models \psi$ and $r \in J(\nu) - \tau_0$. Consequently $q^0 \models [\![\phi[J - \tau_0]]\!]_\nu$, which proves that $\nu \in F^{\phi}(C)$.

2. if ν verifies the second member of the disjunction, we choose q^0 s.t. $\tau_0 \leq J(\nu)^{\downarrow}$ which is possible since $\nu \in \{\tau_{min}(C) \leq J^{\downarrow}\}$.
   - Now, if $\tau_0 \geq J(\nu)^{\uparrow}$, as we did previously, we consider $r = 0$ and prove the result.
   - Otherwise $\tau_0 < J(\nu)^{\uparrow}$ and then
     - if $\mathsf{firable}\,(C) = \emptyset$, then $\exists \rho \in \pi(q^0),\ \rho = q^0 \xrightarrow{d_0} q^0 + d_0$ with $d_0 = \infty$. Consequently, for $r = J(\nu)^{\uparrow} - \tau_0 > 0$, $\rho^*(r) \models \psi$ and $\forall r' < r,\ \rho^*(r') \models \varphi$ because we remain in the same class C, and $r \in J(\nu) - \tau_0$, thus $\nu \in F^{\phi}(C)$.
     - otherwise, $\exists t \in \mathsf{firable}\,(C)$ s.t. $C' = (M', D') = succ(C, t)$ and moreover, according to the formula, $\nu \in \{\tau_{max}(C') \geq J^{\uparrow}\} \wedge D'_{|Par}$.
       Since $\nu \in D'_{|Par}$, the class $[\![C']\!]_\nu$ is reachable, and so there exists: $\rho \in \pi(q^0)$ and $q^1 \in [\![C']\!]_\nu$ s.t. $\rho = q^0 \xrightarrow{d_0} q^0 + d_0 \xrightarrow{t} q^1 \to \ldots$.

We choose q^1 such that $\tau_1 = \tau_{max}(C')(\nu)$, then $\tau_1 \geq J(\nu)^{\uparrow}$ and $d_0 = \tau_1 - \tau_0$.
We consider now $r = \min(d_0, J(\nu)^{\downarrow} - \tau_0)$. We have $\rho^*(r) \models \psi$ and $\forall r' < r,\ \rho^*(r') \models \varphi$ because we are still in the class C. And naturally by construction $r \in J(\nu) - \tau_0$. So, finally $\nu \in F^{\phi}(C)$.

3. in the last case ν verifies the third member of the disjunction, then there exists $t \in \mathsf{firable}\,(()\,C)$ s.t. $\nu \in F^{\phi}_{EU}(succ(C,t))$. We assume here that the computed formula for the successor is correct, i.e. $\exists q^1 \in [\![C']\!]_\nu$, $\exists \rho' \in \pi(q^1)$, $\exists r_1 \in J(\nu) - \tau_1$, s.t. $\rho'^*(r_1) \models \psi$ and $\forall r'_1 < r_1\ \rho'^*(r'_1) \models \varphi$.
In that case there exists $q^0 \in [\![C]\!]_\nu$ and $\rho \in \pi(q^0)$ such that $\rho = q^0 \xrightarrow{d_0 = \tau_1 - \tau_0} q^0 + d_0 \xrightarrow{t} q^1 \rightarrow \rho'$. Then, for $r = d_0 + r_1$ we have $\rho^*(r) \models \psi$ and $\forall r' < r$, $\rho^*(r') \models \varphi$. Besides, $r \leq \tau_1 - \tau_0 + J(\nu)^{\downarrow} - \tau_1 = J(\nu)^{\downarrow} - \tau_0$ and $r \geq \tau_1 - \tau_0 + J(\nu)^{\uparrow} - \tau_1 = J(\nu)^{\uparrow} - \tau_0$, thus $r \in J(\nu) - \tau_0$. So finally $\nu \in F^{\phi}(C)$.

We have proven that when ν verifies the computed formula, in all cases this valuation is correct, and as a result the correctness of the algorithm EU is proven. $\square$

Proof of Theorem 21 (completeness). We consider now a solution of the theoretical problem: $\nu \in F^{\phi}(C)$. We prove that this solution belongs to the set of solutions computed, i.e. $\nu \in F^{\phi}_{EU}(C)$.
We prove this result by induction on the number i of transitions that must be fired to prove the formula ϕ. The induction hypothesis is
$\mathbf{H_i}$: Let be $C = (M, D) \in \mathcal{G}_c(\mathcal{N})$ and $\nu \in D_{|Par}$. If $\exists q^0 \in [\![C]\!]_\nu$, $q^0 \models [\![\phi[J - \tau_0]]\!]_\nu$, i.e. $\exists \rho \in \pi(q^0)$ s.t. $\rho = q^0 \xrightarrow{d_0} q^0 + d_0 \xrightarrow{t_0} q^1 \xrightarrow{d_1} q^1 + d_1 \xrightarrow{t_1} \cdots$ and $\exists r \in J(\nu) - \tau_0$ s.t. $\rho^*(r) \models \psi$ and $\forall r' < r,\ M, \rho^*(r') \models \varphi$, with $r = \sum_{j=0}^{i-1} d_j + \delta$ and $\delta < d_0$, then $\nu \in F^{\phi}_{EU}(C)$.

$\mathbf{H_0}$: we will prove the result for $i = 0$

- Either: $M \models \psi$ and $M \not\models \varphi$. Then, necessarily $r = 0$, and so $0 \in [J(\nu)^{\uparrow} - \tau_0, J(\nu)^{\downarrow} - \tau_0]$. Besides, $\tau_0 \geq \tau_{min}(C)(\nu)$, which proves that $\tau_{min}(C)(\nu) \leq J(\nu)^{\downarrow}$ and thus that $\nu \in \{\tau_{min}(C) \leq J^{\downarrow}\}$. Similarly $\tau_0 \leq \tau_{max}(C)(\nu)$, and so $\nu \in \{\tau_{max}(C) \geq J^{\uparrow}\}$. Then, we have proven that ν belongs to the first member of the disjunctive formula of $F^{\phi}_{EU}(C)$.
- Otherwise: $M \models \psi$ and $M \models \varphi$, and there exists $\rho = q^0 \xrightarrow{d_0} q^0 + d_0 \rightarrow \ldots$, and $0 \leq r < d_0$ s.t. $\rho^*(r) \models \psi$. In that case:
  - if $\theta_0 = \infty$, either $\mathsf{firable}\,(C) = \emptyset$ and since $\tau_{min}(C)(\nu) \leq \tau_0 \leq r + \tau_0 \leq J(\nu)^{\downarrow}$, we directly prove that $\nu \in F^{\phi}_{EU}(C)$. Or it means that $\forall t \in \mathsf{firable}\,(C)\,, J_s(t)^{\downarrow} = \infty$. Then, necessarily $\exists t \in \mathsf{firable}\,(C)$ s.t. $C' =$

$(M', D') = succ(C, t)$ and $\nu \in D'_{|Par}$. Consequently, $\tau_{max}(C') = \infty$, and so $\nu \in F^{\phi}_{EU}(C)$.

- Otherwise, $\exists t$ s.t. $\rho = q^0 \xrightarrow{d_0} q^0 + d_0 \xrightarrow{t} q^1 \rightarrow \cdots$, and thus $t \in \mathsf{firable}(C)$, and for $C' = succ(C, t)$ we have $\nu \in D'_{|Par}$. In that case $\tau_{max}(C')(\nu) \geq \tau_1$ and $\tau_1 = \tau_0 + d_0$. As a result $\tau_{max}(C')(\nu) \geq \tau_0 + d_0 > \tau_0 + r$. Finally, since $r \in J(\nu) - \tau_0$ we get that $\tau_{max}(C')(\nu) \geq J(\nu)^{\uparrow}$ and consequently that $\nu \in F^{\phi}_{EU}(C)$.

We have proven the induction hypothesis for $i = 0$. We now prove the induction.

$\mathbf{H_i} \Rightarrow \mathbf{H_{i+1}}$: we start with the assumptions of H_{i+1}, i.e. $\exists q^0 \in [\![C]\!]_\nu$, $\exists \rho \in \pi(q^0)$, $\exists r \in J - \tau_0$ s.t. $\rho^*(r) \models \psi$ and $\forall r' < r,\ \rho^*(r') \models \varphi$, with $r = \sum_{j=0}^{i} d_j + \delta$.

Since $i + 1 > 0$, $\exists t \in \mathsf{firable}(C)$ s.t. $\rho = q^0 \overset{d_0}{\rightarrow} q^0 + d_0 \overset{t}{\rightarrow} q^1 \rightarrow \ldots$. Let be $C' = (M', D') = succ(C, t)$. $\nu \in D'_{|Par}$ since then the class is reachable. We have $q^1 \in [\![C']\!]_\nu$ and $\tau_1 = \tau_0 + d_0$.
We now consider the sub-sequence of ρ starting from $q^1 : \rho' = q^1 \rightarrow \ldots$. Let be $r_1 = r - d_0 = \sum_{j=1}^{i} d_j + \delta$. Since $r \in J(\nu) - \tau_0$, it follows that $r_1 \in J(\nu) - \tau_1$. Moreover, $\rho'^*(r_1) = \rho^*(r)$, and so $\rho'^*(r_1) \models \psi$. Similarly $\forall r'_1 < r_1\ \rho'^*(r'_1) \models \varphi$. We have just verified all the assumptions for H_i, and by induction we deduce that $\nu \in F^{\phi}_{EU}(C')$.
Then we can deduce that ν verifies the last member of the disjunction in $F^{\phi}_{EU}(C)$. This proves the induction and the completeness of the algorithm. ◻

A.2 Algorithm AU

Theorem 22. *Let $\mathcal{N}$ be a PITPN and $\phi = \forall\varphi\mathcal{U}_J\psi$ a PTCTL formula, where $J \in \mathcal{J}(Par)$. Given an extended parametric state class $C = (M, D)$ of $\mathcal{G}_c(\mathcal{N})$, the correctness and completeness of the semi-algorithm AU are expressed by:*

$$F^{\phi}(C) = F^{\phi}_{AU}(C)$$

Proof of Theorem 22 (correctness). We consider a valuation that belongs to the computed logical formula: $\nu \in F^{\phi}_{AU}(C)$. We prove that this valuation is correct, which means that: $\nu \in F^{\phi}(C) = \{\nu \in D_{|Par} \mid \forall q^0 \in [\![C]\!]_\nu,\ q^0 \models [\![\phi[J - \tau_0]]\!]_\nu\}$.

We directly deduce that $\nu \in D_{|Par}$, it implies that $[\![C]\!]_\nu \neq \emptyset$. And we note that $\nu \in \{\tau_{max}(C) \leq J^{\downarrow}\}$. Then, there is 3 ways to verify the disjunctive formula:

1. If ν verifies the first member of the disjunction (i.e. $M \models \psi\ \wedge \{\tau_{min}(C) \geq J^{\uparrow}\}$).
   Then, $\forall q^0 \in [\![C]\!]_\nu,\ \tau_0 \in J(\nu)$, because, on the one hand $\tau_0 \geq \tau_{min}(C)(\nu) \geq J(\nu)^{\uparrow}$, and on the other hand $\tau_0 \leq \tau_{max}(C)(\nu) \leq J(\nu)^{\downarrow}$. Consequently $\forall \rho \in \pi(q^0)$, for $r = 0$, $\rho^*(r) \models \psi$ and $r \in J(\nu) - \tau_0$. As a result $\nu \in F^{\phi}(C)$.

2. If ν verifies the second member of the disjunction, $\forall q^0 \in [\![C]\!]_\nu$, like previously $\tau_0 \leq J(\nu)^\downarrow$. Then, either $\tau_0 \geq J(\nu)^\uparrow$, in which case the previous proof applies, or $J(\nu)^\uparrow - \tau_0 > 0$ and next:

   – if $\mathsf{firable}\,(C) = \emptyset$ then $\forall \rho \in \pi(q^0)$, $\rho = q^0 \xrightarrow{d_0} q^0 + d_0$ with $d_0 = \infty$. Thus, for $r = J(\nu)^\uparrow - \tau_0 > 0$, we prove the result.

   – Otherwise, $\forall \rho \in \pi(q^0)$. If $\rho = q^0 \xrightarrow{d_0} q^0 + d_0$ with $d_0 = \infty$, as before, we consider $r = J(\nu)^\uparrow - \tau_0$. Else, $\exists t \in \mathsf{firable}\,(C)$ s.t. $\rho = q^0 \xrightarrow{d_0} q^0 + d_0 \xrightarrow{t} q^1 \rightarrow \ldots$ and $d_0 = \tau_1 - \tau_0$. Let be $C' = (M', D') = succ(C, t)$, and $D'' = D' \wedge \{\theta_c > -J^\uparrow\}$ and $C'' = (M', D'')$. From the logical formula we deduce that $\nu \in (F^\phi_{AU}(C'') \vee \neg D''_{|Par})$ and $\nu \in D'_{|Par}$ since in that case the class $[\![C']\!]_\nu$ is reachable.

     - If $\tau_1 \geq J(\nu)^\uparrow$, then for $r = \min(d_0, J(\nu)^\downarrow - \tau_0)$, $\rho^*(r) \models \psi$, $\forall r' < r, \rho^*(r') \models \phi$ since we remain in the class C. Besides $r \in J(\nu) - \tau_0$ since either $r = J(\nu)^\downarrow - \tau_0$, or $r = d_0$ and then $r \leq J(\nu)^\downarrow - \tau_0$ and $r = \tau_1 - \tau_0 \geq J(\nu)^\uparrow - \tau_0$.

     - Otherwise, the last case is when $\tau_1 < J(\nu)^\uparrow$. Then $\exists \nu'$ s.t. $(\nu|\nu') \in (D' \wedge \{\tau_c < J^\uparrow\})$, which means that $(\nu|\nu') \in D''$ or also that $\nu \in D''_{|Par}$. Consequently, we can deduce by elimination from the formula that $\nu \in F^\phi_{AU}(C'')$. By assuming that this result is correct, we get that $\forall q'^1 \in [\![C'']\!]_\nu$, $\forall \rho' \in \pi(q'^1)$, $\exists r_1 \in J(\nu) - \tau'_1$ s.t. $\rho'^*(r_1) \models \psi$, $\forall r'_1 < r_1\ \rho'^*(r'_1) \models \varphi$. This is true in particular for $q^1 \in [\![C'']\!]_\nu$. Now, for $r = d_0 + r_1$ we prove the result for ρ.

3. Finally, the last case is when ν verifies the third member of the disjunctive formula. Then $\forall q^0 \in [\![C]\!]_\nu$, $\forall \rho \in \pi(q^0)$

   – If $\rho = q^0 \xrightarrow{d_0} q^0 + d_0$ with $d_0 = \infty$. As $\mathsf{firable}\,(C) \neq \emptyset$, it means that $\forall t \in \mathsf{firable}\,(C)\,, J_s(t)^\downarrow = \infty$. Then, let be $t \in \mathsf{firable}\,(C)$ and $C' = succ(C, t)$. Since in that case the successors are necessary reachable $\nu \in D'_{|Par}$, and so we should have $\nu \in F^\phi_{AU}(C')$. Or this is impossible since $\tau_{max}(C') = \infty$.

   – Consequently, $\exists t \in \mathsf{firable}\,(C)$ s.t. $\rho = q^0 \xrightarrow{d_0} q^0 + d_0 \xrightarrow{t} q^1 \rightarrow \ldots$. It means that the class $[\![C']\!]_\nu$ is reachable, and thus that $\nu \in D'_{|Par}$. Then, by deduction we get that $\nu \in F^\phi_{AU}(C')$. As already done, by assuming the result correct we show that there exists r_1 such that for $r = d_0 + r_1$ we prove finally that $\nu \in F^\phi(C)$.

We have here proven that in all cases the algorithm is correct. □

Proof of Theorem 22 (completeness). We consider now a solution of the theoretical problem: $\nu \in F^{\phi}(C)$. We prove that this solution belongs to the set of solutions computed, i.e. $\nu \in F^{\phi}_{AU}(C)$.
We prove this result by induction on the maximum number n of transitions that are fired to prove the formula ϕ. The induction hypothesis is
$\mathbf{H_n}$: Let $C = (M, D) \in \mathcal{G}_c(\mathcal{N})$ and $\nu \in D_{|Par}$. If $\forall q^0 \in [\![C]\!]_\nu,\ q^0 \models [\![\phi[J - \tau_0]]\!]_\nu$, i.e. $\forall \rho \in \pi(q^0)$ s.t. $\rho = q^0 \xrightarrow{d_0} q^0 + d_0 \xrightarrow{t_0} q^1 \xrightarrow{d_1} q^1 + d_1 \xrightarrow{t_1} \cdots$, $\exists r \in J(\nu) - \tau_0$ s.t. $\rho^*(r) \models \psi$ and $\forall r' < r,\ \rho^*(r') \models \varphi$, with $r = \sum_{j=0}^{i-1} d_j + \delta$ and $\delta < d_0$, and the number of fired transitions i is such that $i \leq n$, then $\nu \in F^{\phi}_{AU}(C)$.

$\mathbf{H_0}$: we will prove the hypothesis for $n = 0$ Since we have $\nu \in F^{\phi}(C)$ and no transitions are fired it implies that $M \models \psi$. Then

- if $M \not\models \varphi$ we have inevitably $r = 0$, and since $r \in J(\nu) - \tau_0$ it implies that $J(\nu)^{\uparrow} \leq \tau_0 \leq J(\nu)^{\downarrow}$. This is valid $\forall q^0 \in [\![C]\!]_\nu$. Particularly for $q^0 \in [\![C]\!]_\nu$ such that $\tau_0 = \tau_{min}(C)(\nu)$. Thus we prove that $\nu \in \{\tau_{min}(C) \geq J^{\uparrow}\}$. Similarly, for $q^0 \in [\![C]\!]_\nu$ such that $\tau_0 = \tau_{max}(C)(\nu)$ we prove $\nu \in \{\tau_{max}(C) \leq J^{\downarrow}\}$. As result ν belongs to the first member of the disjunctive formula in $F^{\phi}_{AU}(C)$.
- otherwise $M \models \varphi$
  - if $\mathsf{firable}\,(C) = \emptyset$, then as previously for $q^0 \in [\![C]\!]_\nu$ such that $\tau_0 = \tau_{max}(C)(\nu)$, we have $0 \leq r \leq J(\nu)^{\downarrow} - \tau_0 \Rightarrow \tau_0 \leq J(\nu)^{\downarrow}$. And so we prove that $\nu \in \{\tau_{max}(C) \leq J(\nu)^{\downarrow}\}$. This is enough to verify the second member of the disjunctive formula.
  - otherwise, $\forall q^0 \in [\![C]\!]_\nu, \forall t \in \mathsf{firable}\,(C)$ and $\forall \rho \in \pi(q^0)$ s.t. $\rho = q^0 \xrightarrow{d_0 = \tau_1 - \tau_0} q^0 + d_0 \xrightarrow{t} q^1 \rightarrow \ldots$, $\exists r < d_0$ that proves ψ. Let be $C' = succ(C, t)$ and $D'' = D' \wedge \{\tau_c < J^{\uparrow}\}$. $\forall q_1 \in [\![C']\!]_\nu$, since $J(\nu) - \tau_0 \leq r < \tau_1 - \tau_0$ we get that $J(\nu)^{\uparrow} < \tau_1$ and consequently that $[\![D'']\!]_\nu = \emptyset$, which means that $\nu \in \neg D''_{|Par}$. In that case also ν verifies the second disjunction of $F^{\phi}_{AU}(C)$.

Then, we have proven the induction hypothesis for $n = 0$. We now prove the induction.

$\mathbf{H_n} \Rightarrow \mathbf{H_{n+1}}$: We assume the hypothesis of H_{n+1} i.e. $\forall q^0 \in [\![C]\!]_\nu$, $\forall \rho \in \pi(q^0)$, $\exists r \in J - \tau_0$ s.t. $\rho^*(r) \models \psi$ and $\forall r' < r,\ \rho^*(r') \models \varphi$, with $r = \sum_{j=0}^{i-1} d_j + \delta$ and $i \leq n + 1$.
Inevitably, $0 \leq r \leq J(\nu)^{\downarrow} - \tau_0 \Rightarrow \tau_0 \leq J(\nu)^{\downarrow}$ and so for $\tau_0 = \tau_{max}(C)(\nu)$ we prove that $\nu \in \{\tau_{max}(C) \leq J^{\downarrow}\}$. Then

- if $M \not\models \psi$ then $i > 0$, and so $\mathsf{firable}\,(C) \neq \emptyset$. $\forall t \in \mathsf{firable}\,(C)$, let be $C' = succ(C, t)$.

- if $\nu \notin D'_{|Par}$ then $\nu \in \neg D'_{|Par}$.
- else $\forall q^1 \in [\![C']\!]_\nu$, $\forall \rho' \in \pi(q^1)$, $\exists \rho \in \pi(q^0)$ s.t. $\rho = q^0 \xrightarrow{d_0} q^0 + d_0 \xrightarrow{t} q^1 \rightarrow \dots$, and from the initial assumptions $\exists r = \sum_{j=0}^{i-1} d_j + \delta$ that proves ψ. At that point, let be $r_1 = r - d_0$, $r_1 = \sum_{j=1}^{i-1} d_j + \delta$. $r1 \geq 0$ since $r \geq d_0$. Moreover $r_1 = r - d_0 \leq J(\nu)^{\downarrow} - \tau_0 - d_0 = J(\nu)^{\downarrow} - \tau_1$ and $r_1 \geq J(\nu)^{\uparrow} - \tau_0 - d_0 = J(\nu)^{\uparrow} - \tau_1$, which proves that $r_1 \in J(\nu) - \tau_1$. Naturally, $\rho'^{*}(r_1) \models \psi$ and $\forall r'_1 < r_1$, $\rho'^{*}(r'_1) \models \varphi$. Finally, the number of fired transitions is $i - 1 \leq n$. We have checked all the hypothesis for H_n and consequently by induction we get that $\nu \in F^{\phi}_{AU}(C')$.

In summary, for each firable transition, ν verifies the last member of the disjunctive formula in $F^{\phi}_{AU}(C)$.

– otherwise $M \models \psi$. If $\mathsf{firable}(C) = \emptyset$ we directly get that $\nu \in F^{\phi}_{AU}(C)$. Else $\forall t \in \mathsf{firable}(C)$, let be $C' = succ(C, t)$, and $D'' = D' \wedge \{\theta_c \ > \ -J^{\uparrow}\}$ and $C'' = (M', D'')$.
  - if $\nu \notin D''_{|Par}$ then $\nu \in \neg D''_{|Par}$.
  - else $\nu \in D''_{|Par}$ and then $\forall q^1 \in [\![C'']\!]_\nu$, $\forall \rho' \in \pi(q^1)$, $\exists \rho \in \pi(q^0)$ s.t. $\rho = q^0 \xrightarrow{d_0} q^0 + d_0 \xrightarrow{t} q^1 \rightarrow \dots$, and so $\exists r = \sum_{j=0}^{i-1} d_j + \delta$ to prove ψ. As previously, let be $r_1 = r - d_0 = \sum_{j=1}^{i-1} d_j + \delta$. Since $r \in J(\nu) - \tau_0$, $r_1 \in J(\nu) - \tau_1$ and moreover, since $q_1 \in [\![C'']\!]_\nu$ it means that $J(\nu)^{\uparrow} - \tau_1 > 0$, and so that $r_1 > 0$. Then as previously the hypothesis of H_n can be verified and thus by induction we prove that $\nu \in F^{\phi}_{AU}(C'')$.

  In that other case, ν verifies the second disjunctive formula of $F^{\phi}_{AU}(C)$.

As a result, in all cases $\nu \in F^{\phi}_{AU}(C)$ which proves $\mathbf{H_n} \Rightarrow \mathbf{H_{n+1}}$, and then by induction the completeness of the algorithm. □

A.3 Algorithm LT

Theorem 23. *Let $\mathcal{N}$ be a PITPN and $\phi = \varphi \leadsto_{J_r} \psi$ a PTCTL formula, where $J_r \in \mathcal{J}(Par)$ such that $J_r = [0, b]$ with $b \in \mathbb{Q}^+ \cup Par$, or $J_r = [0, \infty[$. Given an extended parametric state class $C = (M, D)$ of $\mathcal{G}_c(\mathcal{N}_{LT})$, the correctness and completeness of the semi-algorithm LT are expressed by:*

$$F^{\phi}(C) = F^{\phi}_{LT}(C)$$

Proof of Theorem 23 (correctness). **Correction:** We consider a valuation that belongs to the computed logical formula: $\nu \in F^{\phi}_{LT}(C)$. We prove that this valuation is correct, which means that: $\nu \in F^{\phi}(C)$.
$\forall q^0 \in [\![C]\!]_\nu, \ \forall \rho \in \pi(q^0)$

1. If $M(P_{LT}) = 0$. Then if $\mathsf{firable}(C) = \emptyset$ or if $\rho = q^0 \xrightarrow{d_0} q^0 + d_0$ with $d_0 = \infty$ we directly prove the result since from this point we always have $M(PLT) = 0$. Else, $\exists t \in \mathsf{firable}(C)$ s.t. $\rho = q^0 \xrightarrow{d_0} q^0 + d_0 \xrightarrow{t} q^1 \rightarrow \dots$. In that case, $\forall 0 \leq r_2 \leq d_0$ we prove the result since $M(PLT) = 0$. For the others, we have $\nu \in F^{\phi}_{LT}(C')$ and besides since $M(PLT) = 0$ the time has been reset in the class C' (and so the two definitions of $F^{\phi}(C')$ are equivalent). So we can induce that $\forall \rho' \in \pi(q_1),\ \forall r'_2 \geq 0,\ \rho'^*(r'_2) \models M(P_{LT}) = 1 \ \Rightarrow\ \exists r'_3 \geq r'_2,$ s.t. $r'_3 - r'_2 \leq J_r(\nu)^{\downarrow}$ and $\rho'^*(r'_3) \models \psi$. Thus, now $\forall r_2 > d_0$ we can take $r'_2 = r_2 - d_0$ and prove the result since then $\rho^*(r_2) = \rho'^*(r'_2)$.

2. Otherwise $M(P_{LT}) = 1$. $\rho = q^0 \xrightarrow{d_0} q^0 + d_0 \rightarrow \dots$
   - If $M \models \psi \wedge\ \tau_{max}(C)(\nu) \leq J_r(\nu)^{\downarrow}$ then we consider $r_1 = 0$ and we have $0 \leq J_r(\nu)^{\downarrow} - \tau_{max}(C)(\nu) \leq J_r(\nu)^{\downarrow} - \tau_0$ and naturally $\rho^*(r_1) \models \psi$. Next, either $\mathsf{firable}(C) = \emptyset$ and so $d_0 = \infty$. In that case $\forall r_2 \geq r_1$ we consider $r_3 = r_2$ and we prove the result for the last states. Or $\nu \in F^{\phi}_{LT}(C')$ and as previously we can induce that the result is also true $\forall r_2 > d_0$ (in that case we only consider the second definition because $M \models \psi \Rightarrow M'(P_{LT}) = 0$).

   - Otherwise, $M \not\models \psi$ and thus $\mathsf{firable}(C) \neq \emptyset$. The case $\rho = q^0 \xrightarrow{d_0} q^0 + d_0$ with $d_0 = \infty$ can be eliminated similarly to the algorithm AU since then for all successors we still have $M'(P_{LT}) = 1$ and with the restriction $\tau_{max}(C') \neq \infty$ we get a contradiction.

   - Finally, $M \not\models \psi$ and $\exists t \in \mathsf{firable}(C)$ s.t. $\rho = q^0 \xrightarrow{d_0 = \tau_1 - \tau_0} q^0 + d_0 \xrightarrow{t} q^1 \rightarrow \dots$. We have $\nu \in F^{\phi}_{LT}(C')$ and $q_1 \in [\![C']\!]_{\nu}$ and since ψ has not been found in C we have $M'(P_{LT}) = 1$.
     By assuming the result of $F^{\phi}_{LT}(C')$ correct, $\forall \rho' \in \pi(q^1)$
     - we first get that $\exists r'_1 \geq 0$ s.t. $r'_1 \leq J_r(\nu) - \tau_1$ and $\rho'^*(r'_1) \models \psi$. Then for $r_1 = r'_1 + d_0$ we have $r_1 \leq J_r(\nu) - \tau_1 + \tau_1 - \tau_0$ and so $r_1 \leq J_r(\nu) - \tau_0$ and $\rho^*(r_1) = \rho'^*(r'_1)$. So we have found psi.

     - for the other states i.e. $\forall r_2 \geq r_1$, we induce that $\forall r'_2 \geq r'_1,\ \rho'^*(r'_2) \models M(P_{LT}) = 1 \ \Rightarrow\ \exists r'_3 \geq r'_2,$ s.t. $r'_3 - r'_2 \leq J_r(\nu)^{\downarrow}$ and $\rho^*(r'_3) \models \psi$. We consider now $r'_2 = r_2 - d_0$. We have $r'_2 \geq r'_1$ and thus $\exists r'_3$. So we now consider $r_3 = r'_3 + d_0$ and we finally prove the result.

We have proven that in both cases the algorithm is correct. □

Proof of Theorem 23 (completeness). We consider now a solution of the theoretical problem: $\nu \in F^{\phi}(C)$. We prove that this solution belongs to the set of solutions computed, i.e. $\nu \in F^{\phi}_{LT}(C)$.
Let consider the two following cases:

1. If $M(P_{LT}) = 0$ then $\nu \in F^{\phi}_{LT}(C)$ implies that $\forall q^0 \in [\![C]\!]_\nu$, $\forall \rho \in \pi(q^0)$, $\forall r_2 \geq 0$ we prove the formula for all states.
   If $\mathsf{firable}(C) = \emptyset$ the result is directly proven. Else, $\forall t \in \mathsf{firable}(C)$, let be $C' = succ(C,t)$. Either $D'_{|Par} = \emptyset$ or $\forall q_1 \in [\![C']\!]_\nu$, there exists $\rho = q^0 \xrightarrow{d_0} q^0 + d_0 \xrightarrow{t} q^1 \rightarrow \ldots$. Now $\forall \rho' \in \pi(q_1)$ and $\forall r'_2 \geq 0$, we get that $\rho'^*(r'_2) = \rho^*(r_2)$. Thus we have verified the second definition for $F_\phi(C')$ which is the only one when $M(P_{LT}) = 0$ and consequently we can induce that $\nu \in F^{\phi}_{LT}(C')$. So we prove that $\nu \in F^{\phi}_{LT}(C)$.

2. Otherwise $M(P_{LT}) = 1$. Then $\exists r_1 \geq 0$ such that we have $\rho^*(r_1) \models \psi$ and $r_1 \leq J_r(\nu)^{\downarrow} - \tau_0$. This is valid for all q^0 in particular for q^0 s.t. $\tau_0 = \tau_{max}(C)(\nu)$, which proves that $\nu \in \{\tau_{max}(C) \leq J^{\downarrow}_r\}$.
   Then, if $\mathsf{firable}(C) = \emptyset$ necessarily $M \models \psi$ (since then $\rho^*(r_1) \in C$), which proves $\nu \in F^{\phi}_{LT}(C)$.
   Otherwise, $\forall t \in \mathsf{firable}(C)$, let be $C' = succ(C,t)$. Either $D'_{|Par} = \emptyset$ or $\forall q_1 \in [\![C']\!]_\nu$, there exists $\rho = q^0 \xrightarrow{d_0} q^0 + d_0 \xrightarrow{t} q^1 \rightarrow \ldots$. Now $\forall \rho' \in \pi(q_1)$
   - if $r_1 \leq d_0$, then $M \models \psi$ and as we did for $M(P_{LT}) = 0$ we can induce that $\nu \in F^{\phi}_{LT}(C')$ by verifying the second definition.
   - otherwise $r_1 > d_0$. So $\exists r'_1 = r_1 - d_0$ such that $r'_1 \leq J_r(\nu) - \tau_0 - (\tau_1 - \tau_0) \leq J_r(\nu) - \tau_1$ and naturally $\rho'^*(r'_1) \models \psi$. Next, from the definition of $F^{\phi}(C)$, $\forall r'_2 \geq r'_1$ we can take $r_2 = r'_2 + d_0$ such that $\rho^*(r_2) = \rho'^*(r'_2)$. Again this proves $\nu \in F^{\phi}(C')$ and by induction that $\nu \in F^{\phi}_{LT}(C')$.

   So we prove that $\nu \in F^{\phi}_{LT}(C)$.

This two cases prove the completeness of the semi-algorithm. $\square$