Symbolic Unfolding of Parametric Stopwatch Petri Nets

L.-M. Traonouez[1], B. Grabiec[2], C. Jard[2], D. Lime[3] and O. H. Roux[3]*

[1] Università di Firenze, Dipartimento di Sistemi e Informatica, Italy

[2] ENS Cachan & INRIA, IRISA, Rennes, France
Université européenne de Bretagne

[3] École Centrale de Nantes & Université de Nantes, IRCCyN, Nantes, France

Abstract. This paper proposes a new method to compute symbolic unfoldings for safe Stopwatch Petri Nets (SwPNs), extended with time parameters, that symbolically handle both the time and the parameters. We propose a concurrent semantics for (parametric) SwPNs in terms of timed processes *à la* Aura and Lilius. We then show how to compute a symbolic unfolding for such nets, as well as, for the subclass of safe time Petri nets, how to compute a finite complete prefix of this unfolding.
Our contribution is threefold: unfolding in the presence of stopwatches or parameters has never been addressed before. Also in the case of time Petri nets, the proposed unfolding has no duplication of transitions and does not require read arcs and as such its computation is more local. Finally the unfolding method is implemented (for time Petri nets) in the tool Romeo.

Keywords: unfolding, time Petri nets, stopwatches, parameters, symbolic methods

1 Introduction

The analysis of concurrent systems is one of the most challenging practical problems in computer science. Formal specification using Petri nets has the advantage to focus on the tricky part of such systems, that is parallelism, synchronization, conflicts and timing aspects. Among the different analysis techniques, we chose to develop the work on unfoldings [9].

Unfoldings were introduced in the early 1980s as a mathematical model of causality and became popular in the domain of computer aided verification. The main reason was to speed up the standard model-checking technique based on the computation of the interleavings of actions, leading to a very large state space in case of highly concurrent systems. The seminal papers are [14] and [8]. They dealt with basic bounded Petri nets.

Since then, the technique has attracted more attention, and the notion of unfolding has been extended to more expressive classes of Petri nets (Petri nets

* This work was partially funded by the ANR national research program DOTS (ANR-06-SETI-003).

with read and inhibitor arcs [7,3], unbounded nets [1], high-level nets [12], and time Petri nets [6]).

Advancing this line of works, we present in this paper a method to unfold safe parametric stopwatch Petri nets. Stopwatch Petri nets (SwPNs) [5] are a strict extension of the classical time Petri nets *à la* Merlin (TPNs) [15,4] and provide a means to model the suspension and resumption of actions with a memory of the "work" done before the suspension. This is very useful to model real-time preemptive scheduling policies for example.

The contribution of this paper is a new unfolding algorithm addressing the problem for stopwatch and parametric models for the first time. When applied to the subclass of time Petri nets, it provides an alternative to [6] and improves on the latter method by providing a more compact unfolding and not requiring read arcs in the unfolding (if the TPN itself has no read arcs of course). We also provide a way to compute a finite complete prefix of the unfolding for (safe) TPNs. Note this is the best we can do as most interesting properties, such as reachability, are undecidable in time Petri nets in presence of stopwatches [5] or parameters [16].

While not extremely difficult from a theoretical point of view, we think that the handling of parameters is of utmost practical importance: adding parameters in specifications is a real need. It is often difficult to set them a priori: indeed, we expect from the analysis some useful information about their possible values. This feature of genericity clearly adds some "robustness" to the modeling phase. It is important to note that, as for time, we handle these parameters symbolically to achieve this genericity and the unfolding technique synthesizes all their possible values as linear constraint expressions.

Finally, note that the lack of existence of a finite prefix in the stopwatch or parametric cases is not necessarily prohibitive as several analysis techniques, such as supervision, can do without it [10]. Practical experience also demonstrates that even for very expressive models, such as Linear Hybrid Automata [11], the undecidability of the interesting problems still allows to analyze them in many cases.

Organization of the paper. Section 2 gives preliminary definitions and Section 3 propose an unfolding method of stopwatch parametric Petri nets based on an original way of determining conflicts in the net. Section 4 shows how to compute a complete finite prefix of the unfolding of a time Petri net. Finally in Section 5, we discuss open problems and future work.

2 Definitions

We denote by $\mathbb{N}$ the set of non-negative integers, by $\mathbb{Q}$ the set of rational numbers and $\mathbb{R}$ the set of real numbers. For $A \in \{\mathbb{Q}, \mathbb{R}\}$, $A_{\geq 0}$ (resp. $A_{>0}$) denotes the subset of non-negative (resp. strictly positive) elements of A. Given $a, b \in \mathbb{N}$ such that $a \leq b$, we denote by $[a..b]$ the set of integers greater or equal to a and less or equal to b. For any set X, we denote by $|X|$ its cardinality.

For a function f on a domain D and a subset C of D, we denote by $f_{|C}$ the restriction of f to C.

Let X be a finite set. A (rational) *valuation* of X is a function from X to $\mathbb{Q}$. A (rational) *linear expression* on X is an expression of the form $a_1x_1 + \cdots + a_nx_n$, with $n \in \mathbb{N}$, $\forall i, a_i \in \mathbb{Q}$ and $x_i \in X$. A *linear constraint* on X is an expression of the form $L_X \sim b$, where L_X is a linear expression on X, $b \in \mathbb{Q}$ and $\sim \in \{<, \leq, \geq, >\}$. Given a linear expression $L = a_1x_1 + \cdots + a_nx_n$ on X and a rational valuation v on X, we denote $v(L)$ the rational number $a_1v(x_1) + \cdots + a_nv(x_n)$. Similarly for a linear constraint $C = L \sim b$, we note $v(C)$ the Boolean expression $(v(L) \sim b)$. We extend this notation in the same way for conjunctions, disjunctions and negations of constraints.

For the sake of readability, when non-ambiguous, we will "flatten" nested tuples, *e.g.* $\langle\langle\langle B, E, F\rangle, l\rangle, v, \theta\rangle$ will be written $\langle B, E, F, l, v, \theta\rangle$.

2.1 Unfolding Petri nets

Definition 1 (Place/transition net). *A place/transition net with read arcs (P/T net) is a tuple $\langle P, T, W, W_r\rangle$ where: P is a finite set of* places*, T is a finite set of* transitions*, with $P \cap T = \emptyset$, $W \subseteq (P \times T) \cup (T \times P)$ is the* transition incidence relation *and $W_r \subseteq P \times T$ is the* read incidence relation

This structure defines a directed bipartite graph such that $(x, y) \in W \cup W_r$ iff there is an arc from x to y.

We further define, for all $x \in P \cup T$, the following sets: ${}^\bullet x = \{y \in P \cup T \mid (y, x) \in W\}$, ${}^\diamond x = \{y \in P \cup T \mid (y, x) \in W_r\}$ and $x^\bullet = \{y \in P \cup T \mid (x, y) \in W\}$. These set definitions naturally extend by union to subsets of $P \cup T$.

A *marking* $m : P \to \mathbb{N}$ is a function such that (P, m) is a multiset. For all $p \in P$, $m(p)$ is the number of *tokens* in the place p. In this paper we restrict our study to *1-safe* nets, *i.e.* nets such that $\forall p \in P,\ m(p) \leq 1$. Therefore, in the rest of the paper, we will usually identify the marking m with the set of places p such that $m(p) = 1$. In the sequel we will call *Petri net* (with read arcs) a marked P/T net, *i.e.* a pair $\langle \mathcal{N}, m\rangle$ where $\mathcal{N}$ is a P/T net and m a marking of $\mathcal{N}$, called *initial marking.*

A transition $t \in T$ is said to be enabled by the marking m if ${}^\bullet t \cup {}^\diamond t \subseteq m$. We denote by $\mathsf{en}(m)$, the set of transitions enabled by m.

2.2 Semantics of true concurrency

There is a path $x_1, x_2, \ldots, x_n$ in a P/T net iff $\forall i \in [1..n]$, $x_i \in P \cup T$ and $\forall i \in [1..n-1], (x_i, x_{i+1}) \in W \cup W_r$.

In a P/T net, consider $x, y \in P \cup T$. x and y are *causally related*, which we denote by $x < y$, iff there exists a path in the net from x to y. The causal past of a transition t is called *local configuration* and denoted by $\lceil t \rceil$, and is constituted by the transitions that causally precede t, *i.e.* $\lceil t \rceil = \{t' \in T \mid t' < t\}$.

The addition of the read arcs introduces another causal relation between two transitions $x, y \in T$, that is called *weak causality* and denoted by $x \nearrow y$, iff $x < y \vee {}^\diamond x \cap {}^\bullet y \neq \emptyset$. This notion is already presented in [7]. The relation denotes that the firing of the transition x happens before the one of y.

The two causal relations induce a relation of conflicts between the transitions of the net. A set $X \subseteq T$ of transitions are said to be in conflict, noted $\#X$, when some transitions consumed the same token, or when the weak causality defines a cycle in this set. Formally:

$$\#X = \begin{cases} \exists x, y \in X \ : \ x \neq y \wedge {}^\bullet x \cap {}^\bullet y \neq \emptyset \ \vee \\ \exists x_0, x_1, \ldots, x_n \in X \ : \ x_0 \nearrow x_1 \nearrow \ldots x_n \nearrow x_0 \end{cases}$$

Definition 2 (Occurrence net). *An* occurrence net *is an acyclic P/T net* $\langle B, E, F, F_r \rangle$*:*

- *finite by precedence ($\forall e \in E,\ \lceil e \rceil$ is finite),*
- *such that each place has at most one input transition ($\forall b \in B,\ |{}^\bullet b| \leq 1$),*
- *and such that there is no conflicts in the causal past of each transition ($\forall e \in E,\ \neg\#\{e \cup \lceil e \rceil\}$).*

We use the classical terminology of *conditions* and *events* to refer to the places B and the transitions E in an occurrence net.

Definition 3 (Branching process). *A* branching process *of a Petri net* $\mathcal{N} = \langle P, T, W, W_r, m_0 \rangle$ *is a labeled occurrence net* $\beta = \langle \mathcal{O}, l \rangle$ *where* $\mathcal{O} = \langle B, E, F, F_r \rangle$ *is an occurrence net and* $l : B \cup E \to P \cup T$ *is the labeling function such that:*

- $l(B) \subseteq P$ *and* $l(E) \subseteq T$,
- *for all $e \in E$, the restriction $l_{|{}^\bullet e}$ of l to ${}^\bullet e$ is a bijection between ${}^\bullet e$ and ${}^\bullet l(e)$,*
- *for all $e \in E$, the restriction $l_{|{}^\diamond e}$ of l to ${}^\diamond e$ is a bijection between ${}^\diamond e$ and ${}^\diamond l(e)$,*
- *for all $e \in E$, the restriction $l_{|e^\bullet}$ of l to $e^\bullet$ is a bijection between $e^\bullet$ and $l(e)^\bullet$,*
- *for all $e_1, e_2 \in E$, if ${}^\bullet e_1 = {}^\bullet e_2$, ${}^\diamond e_1 = {}^\diamond e_2$ and $l(e_1) = l(e_2)$ then $e_1 = e_2$.*

E should also contain the special event $\perp$, such that: ${}^\bullet\perp = \emptyset$, ${}^\diamond\perp = \emptyset$, $l(\perp) = \emptyset$, and $l_{|\perp^\bullet}$ is a bijection between $\perp^\bullet$ and m_0.

Branching processes can be partially ordered by a *prefix relation*. For example, the process $\{e_1, e_2, e_3\}$ is a prefix of the branching process in Fig. 1b in which t_1 is fired only once. There exists the greatest branching process according to this relation for any Petri net $\mathcal{N}$, which is called the *unfolding* of $\mathcal{N}$. Let $\beta = \langle B, E, F, F_r, l \rangle$ be a branching process.

A co-*set* in β is a set $B' \subseteq B$ of conditions that are in concurrence, that is to say without causal relation or conflict, *i.e.* $\forall b, b' \in B', \neg(b < b')$ and $\neg\# \bigcup_{b \in B'} ({}^\bullet b \cup \lceil {}^\bullet b \rceil)$.

A *configuration* of β is a set of events $E' \subseteq E$ which is causally closed and conflict-free, that is to say $\forall e' \in E', \forall e \in E,\ e < e' \Rightarrow e \in E'$ and $\neg\#E'$. In particular the local configuration $\lceil e \rceil$ of an event e is a configuration.

A *cut* is a maximal co-set (inclusion-wise). For any configuration E', we can define the cut $\mathsf{Cut}(E') = E'^\bullet \setminus {}^\bullet E'$, which is the marking of the Petri net obtained after executing the sequence of events in E'.

An *extension* of β is a pair $\langle t, e \rangle$ such that e is an event not in E, ${}^\bullet e \cup {}^\diamond e \subseteq B$ is a co-set, the restriction of l to ${}^\bullet e$ is bijection between ${}^\bullet e$ and ${}^\bullet t$, the restriction of l to ${}^\diamond e$ is bijection between ${}^\diamond e$ and ${}^\diamond t$, and there is no $e' \in E$ s.t. $l(e') = t$, ${}^\bullet e' = {}^\bullet e$ and ${}^\diamond e' = {}^\diamond e$. Adding e to E and labeling e with t gives a new branching process.

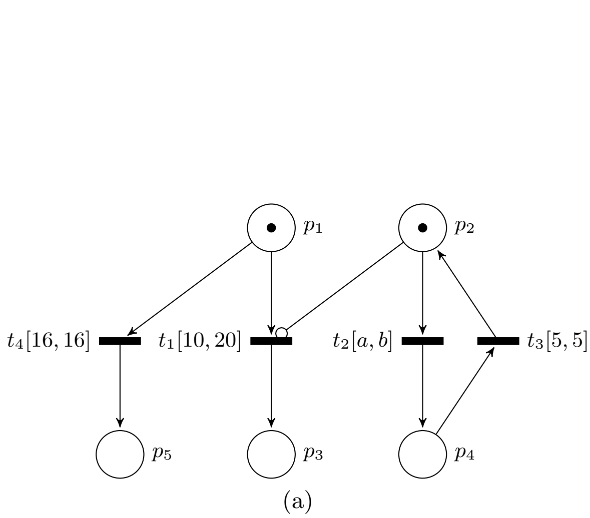

Fig. 1. A parametric stopwatch Petri net (a) and a branching processes of its underlying (untimed) Petri net (b). Stopwatch arcs are drawn with a circle tip and read arcs with a diamond tip.

Example 1. Fig. 1b shows a branching process obtained by unfolding the net presented in Fig. 1a (ignoring any timing or parameter information). The labels are figured inside the nodes. The branching process in Fig. 1b includes two firings of t_1 after executing the loop t_2, t_3. It could be repeated infinitely many times, leading to an infinite unfolding.

2.3 Stopwatch Petri nets

A mainstream way of adding time to Petri nets is by equipping transitions with a time interval. This model is known as Time Petri nets (TPNs) [15,4]. We use a further extension of TPNs featuring stopwatches, called Stopwatch Petri nets (SwPNs) and originally proposed in [5]. Stopwatches allow the modelling of suspension / resumption of actions, which has many useful applications like modelling real-time preemptive scheduling policies [13].

The added expressivity comes at the expense of decidability: most interesting problems, such as reachability, liveness, etc. are undecidable for SwTPNs, even when bounded [5]. They are decidable however when restricting to bounded TPNs [4].

Definition 4 (Stopwatch Petri net). *A Stopwatch Petri net (with read arcs) SwPN is a tuple* $\langle P, T, W, W_r, W_s, m_0, \mathsf{eft}, \mathsf{lft}\rangle$ *where:* $\langle P, T, W, W_r, m_0\rangle$ *is a Petri net,* $W_s \subseteq P \times T$ *is the* stopwatch incidence relation*, and* $\mathsf{eft} : T \to \mathbb{Q}_{\geq 0}$ *and* $\mathsf{lft} : T \to \mathbb{Q}_{\geq 0} \cup \{\infty\}$ *are functions satisfying* $\forall t \in T,\ \mathsf{eft}(t) \leq \mathsf{lft}(t)$*, and respectively called* earliest (eft) *and* latest (lft) *transition firing times.*

Given a SwPN $\mathcal{N} = \langle P, T, W, W_r, W_s, m_0, \mathsf{eft}, \mathsf{lft} \rangle$, we denote by $\mathsf{Untimed}(\mathcal{N})$ the Petri net $\langle P, T, W, W_r \cup W_s, m_0 \rangle$. Note that in $\mathsf{Untimed}(\mathcal{N})$ stopwatch arcs are transformed into read arcs. For any transition t, we define the set of its *activating* places as ${}^\circ t = \{p \in P \mid (p, t) \in W_s\}$. A transition is said to be *active* in marking M if it is enabled by M and ${}^\circ t \subseteq M$. An enabled transition that is not active is said to be *suspended*.

Intuitively, the semantics of TPN states that any enabled transition measures the time during which it has been enabled and an enabled transition can only fire if that time is within the time interval of the transition. Also, unless it is disabled by the firing of another transition, the transition must fire within the interval: a finite upper bound for the time interval then means that the transition will become urgent at some point. For SwPNs, the time during which the transition has been enabled progresses if and only if all its activating places are marked. Otherwise the stopwatch is "frozen" and keeps its current value.

More formally, we define the concurrent semantics of SwPNs using the time processes of Aura and Lilius [2]. Let us first recall the definition of these time processes:

Definition 5 (Time process). *A* time process *of a Stopwatch Petri net $\mathcal{N}$ is a pair $\langle E', \theta \rangle$, where E' is a configuration of (a branching process of)* $\mathsf{Untimed}(\mathcal{N})$ *and $\theta : E' \to \mathbb{R}_{\geq 0}$ is a timing function giving a firing date for any event of E'.*

Let $\langle E', \theta \rangle$ be a time process of a SwPN $\mathcal{N} = \langle P, T, W, W_r, W_s, m_0, \mathsf{eft}, \mathsf{lft} \rangle$ and $\beta = \langle B, E, F, F_r, l \rangle$ be the associated branching process of $\mathsf{Untimed}(\mathcal{N})$. We note ${}^*e = {}^\bullet e \cup \{b \in {}^\diamond e \mid l(b) \in {}^\diamond l(e)\}$ the set of conditions that enabled an event e in the process E. These conditions are the consumed conditions and the read conditions due to read arcs, but it excludes the read conditions due to stopwatches.

Let $B' \subseteq E'^\bullet$ be a co-set and $t \in T$ be a transition enabled by $l(B')$. We define the *enabling date* of t by B' as: $\mathsf{TOE}(B', t) = \max(\{\theta({}^\bullet b) \mid b \in B' \wedge l(b) \in {}^\bullet t \cup {}^\diamond t\})$. This means that we measure the time during which the transition has been enabled. By extension, for any event e, we note $\mathsf{TOE}(e) = \mathsf{TOE}({}^*e, l(e))$. We also define the set of events temporally preceding an event $e \in E'$ as: $\mathsf{Earlier}(e) = \{e' \in E' \mid \theta(e') < \theta(e)\}$, and we note $C_e = \mathsf{Cut}(\mathsf{Earlier}(e))$.

When dealing with stopwatches, the enabling date is not sufficient to determine the firing dates of the event, and is replaced by the notion of activity duration. For any co-set B', we define its *duration* up to some date θ as:

$$\mathsf{dur}(B', \theta) = \min\{\min_{e \in B'^\bullet}\{\theta(e)\}, \theta\} - \max_{b \in B'}\{\theta({}^\bullet b)\}$$

Then, for a transition t enabled by a co-set B', we define its *active* co*-sets* $\mathsf{Acos}(B', t)$ as all the co-sets A s.t.

- A is in the causal past of B',
- the conditions that enabled t in B also belong to A,
- t is active in A.

Finally the *activity duration* of the transition t at some date θ is:

$$\mathsf{adur}(B,t,\theta) = \sum_{A \in \mathsf{Acos}(B,t)} \mathsf{dur}(A,\theta)$$

By extension, for any event e, we note $\mathsf{Acos}(e) = \mathsf{Acos}({}^*e, l(e))$, and $\mathsf{adur}(e,\theta) = \mathsf{adur}({}^*e, l(e), \theta)$.

The semantics of a Stopwatch Petri net is then defined using the notion of *validity* of time processes.

Definition 6 (Valid time process for SwPNs). *A time process is* valid *iff* $\theta(\bot) = 0$ *and the following constraints are satisfied,* $\forall e \in E'$ $(e \neq \bot)$*:*

$$\theta(e) \geq \max(\{\theta({}^\bullet b) \mid b \in {}^\bullet e \cup {}^\diamond e\}) \tag{1}$$

$$\mathsf{adur}(e, \theta(e)) \geq \mathsf{eft}(l(e)) \tag{2}$$

$$\forall t \in \mathsf{en}(l(C_e)), \mathsf{adur}(C_e, t, \theta(e)) \leq \mathsf{lft}(t) \tag{3}$$

Condition 1 ensures that time progresses. Condition 2 states that to fire a transition $l(e)$ by an event e, it must have been active for at least a duration equal to $\mathsf{eft}(l(e))$ before being fired. Condition 3 states that at the firing date of an event e, the activity duration of no transition t can exceed its maximum firing time $\mathsf{lft}(t)$. Notice that if the former is purely local to the transition t, the latter refers to all enabled transitions in the net, which adds causality between events that are not causally related in the underlying untimed net.

It is easy to see that in the case of TPNs without stopwatches this definition reduces to the definition of Aura and Lilius [2] since, for any transition t enabled by a co-set B, we then have $\mathsf{Acos}(B,t) = B$ and $\forall \theta,\ \mathsf{dur}(B,\theta) = \theta - \mathsf{TOE}(B,t)$.

Note that, in this paper, we consider only Petri nets with non-zeno behavior.

Finally, we extend SwPNs with parameters, a model introduced in [16].

Definition 7 (Parametric Stopwatch Petri net). *A* Parametric *Stopwatch Petri net (PSwPN) is a tuple* $\mathcal{N} = \langle P, T, W, W_r, W_s, m_0, \mathsf{eft}, \mathsf{lft}, \Pi, D_\Pi \rangle$ *where:* $\langle P, T, W, W_r, m_0 \rangle$ *is a Petri net,* W_s *is the stopwatch incidence relation as before,* Π *is a finite set of* parameters $(\Pi \cap (P \cup T) = \emptyset)$*,* D_Π *is a conjunction of linear constraints describing the set of* initial constraints *on the parameters, and* eft *and* lft *are functions on* T *such that for all* $t \in T$*,* $\mathsf{eft}(t)$ *and* $\mathsf{lft}(t)$ *are* rational linear expressions *on* Π *(or* $\mathsf{lft}(t)$ *is infinite).*

Definition 8 (Semantics of a PSwPN). *Let* $\mathcal{N} = \langle P, T, W, W_r, W_s, m_0, \mathsf{eft}, \mathsf{lft}, \Pi, D_\Pi \rangle$*. Given a rational valuation* v *on* Π *such that* $v(D_\Pi)$ *is true, we define the semantics of* $\mathcal{N}$ *as the SwPN* $\mathcal{N}_v = \langle P, T, W, W_r, W_s, m_0, v(\mathsf{eft}), v(\mathsf{lft}) \rangle$*.*

Example 2. Fig. 1a gives an example of a PSwPN. Notice that the time interval of transition t_2 refers to two parameters a and b. The only initial constraint is $D_\Pi = \{a \leq b\}$.

3 Unfolding

The method we propose to unfold parametric stopwatch Petri nets is based on an original way of determining conflicts in the net. In the non parametric timed case (no stopwatch), unfoldings built with this method differ in general from those of [6]. In [6], the emphasis is put on the on-line characteristic of the algorithm: it is a pessimistic approach that ensures that events and constraints put in the unfolding cannot be back into question. This leads possibly to unnecessary duplication of events. In contrast, we propose here an optimistic approach, which requires to dynamically compute the conflicts, and sometimes to backtrack on the constraints.

We propose to refine the conflict notion by defining a relation of direct conflict.

Definition 9 (Direct conflict). *Let $\mathcal{O} = \langle B, E, F, F_r \rangle$ be an occurrence net. Two events $e_1, e_2 \in E$ are in direct conflict, which we denote by $e_1 \ \mathsf{conf}\ e_2$, iff*

$$\begin{cases} \neg\#\{e_2 \cup \lceil e_2 \rceil \cup \lceil e_1 \rceil\} \\ \neg\#\{e_1 \cup \lceil e_1 \rceil \cup \lceil e_2 \rceil\} \\ {}^\bullet e_1 \cap {}^\bullet e_2 \neq \emptyset \end{cases}$$

The first two conditions amount to say that ${}^\bullet e_1 \cup {}^\bullet e_2$ is a co-set. Direct conflicts are central to our study for they are at the root of all conflicts.

Example 3. The branching process presented in Fig. 1b contains direct conflicts $e_1 \ \mathsf{conf}\ e_5$, $e_4 \ \mathsf{conf}\ e_5$ and $e_1 \ \mathsf{conf}\ e_4$. e_1 and e_2 are only weakly ordered ($e_1 \nearrow e_2$).

3.1 Time Branching Processes

We shall now extend the notion of branching process with time information, allowing us to define the symbolic unfolding of PSwPNs. We do this in a way similar to extending configurations to time processes, by adding a function labeling events with their firing date. In a branching process however, some events may be in conflict, which means that some of them may not fire at all. We will account for this situation by labeling an event that never fires with $+\infty$.

The introduction of time in Petri nets reduces the admissible discrete behaviors, but induces new kinds of causal relations. For instance, in the TPN of Fig. 2(a), the firing of t_1 is only possible if t_3 is fired before t_2, which liberates the conflict between t_1 and t_2.

In the unfolding method of TPNs proposed in [6] these relations are handled by using read arcs in the unfolding, so that the firing of an event is duplicated according to the local state in which it is fired. The drawback in this approach is that it can lead to numerous unnecessary duplications of an event. For instance, considering now the TPN of Fig. 2(b), the firing of t_4 is possible in the states (p_1, p_4), (p_2, p_4) or (p_3, p_4), leading to a duplication of the event in each case.

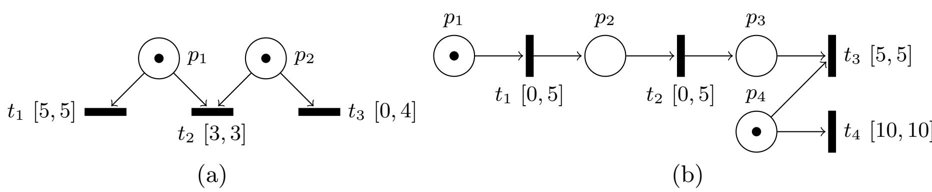

Fig. 2. Time-induced causality in time Petri nets

In our approach we try to express more local conditions by referring only to events in direct conflict. In the example of Fig. 2(b), this is expressed by the relation e_{t_3} conf e_{t_4} that allows the derivation of the constraints on the firing date of these two events. The cost of this approach is that until t_2 has not been fired, no restriction is put on the firing of t_4, and additional constraints are only added afterwards.

Definition 10 (Time branching process). *Given a SwPN $\mathcal{N} = \langle P, T, W, W_r, W_s, m_0, \mathsf{eft}, \mathsf{lft}\rangle$, a* Time Branching Process *(TBP) of $\mathcal{N}$ is a tuple $\langle \beta, \theta\rangle$ where $\beta = \langle B, E, F, F_r, l\rangle$ is a branching process of* $\mathsf{Untimed}(\mathcal{N})$ *and $\theta : E \to \mathbb{R}_{\geq 0} \cup \{\infty\}$ is a timing function giving a firing date for any event in E.*

As for time processes we define the notion of validity of the timing function of time branching process. In the sequel, we will say that a TBP is valid if its timing function is valid.

Definition 11 (Valid timing function for a TBP). *Given a PSwPN $\mathcal{N} = \langle P, T, W, W_r, W_s, m_0, \mathsf{eft}, \mathsf{lft}, \Pi, D_\Pi\rangle$ and a valuation $v \in D_\Pi$ of the parameters, let $\Gamma = \langle B, E, F, F_r, l, \theta\rangle$ be a time branching process of $\mathcal{N}_v$. θ is a* valid *timing function for Γ iff $\theta(\bot) = 0$ and $\forall e \in E$ $(e \neq \bot)$,*

$$\Big[\theta(e) \neq \infty \ \wedge\ \theta(e) \geq \max(\{\theta({}^\bullet b) \mid b \in {}^\bullet e \cup {}^\diamond e\}) \tag{4}$$

$$\wedge\ \mathsf{adur}(e, \theta(e)) \geq v(\mathsf{eft}(l(e))) \tag{5}$$

$$\wedge\ \mathsf{adur}(e, \theta(e)) \leq v(\mathsf{lft}(l(e))) \tag{6}$$

$$\wedge\ \forall e' \in E \ s.t.\ e' \ \mathsf{conf}\ e,\ \theta(e') = \infty \tag{7}$$

$$\wedge\ \forall e' \in E \ s.t.\ e \nearrow e',\ \theta(e) \leq \theta(e')\Big] \tag{8}$$

$$\vee\ \Big[\theta(e) = \infty \ \wedge\ \exists b \in {}^\bullet e,\ \theta({}^\bullet b) = \infty\Big] \tag{9}$$

$$\vee\ \Big[\theta(e) = \infty \ \wedge\ \exists e' \in E \ s.t.\ (e \ \mathsf{conf}\ e' \vee e \nearrow e')$$

$$\wedge\ \theta(e') \neq \infty \wedge \mathsf{adur}(e, \theta(e')) \leq v(\mathsf{lft}(l(e)))\Big] \tag{10}$$

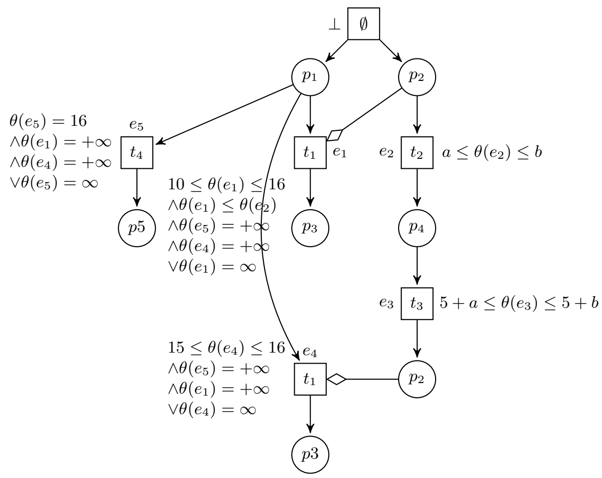

Fig. 3. A TBP with symbolic constraints for the PSwPN of Fig. 1a.

Additionally, if $\exists\{e_0, e_1, \ldots, e_n\} \subseteq E$ *s.t.* $e_0 \nearrow e_1 \nearrow \cdots \nearrow e_n \nearrow e_0$ *then* $\exists i \in [0..n]$ *s.t* $\theta(e_i) = \infty$.
In these constraints, the usual operators are naturally extended to $\mathbb{R}_{\geq 0} \cup \{\infty\}$.

Eq. 4 ensures that time progresses. Eq. 5 constrains the earliest firing date and Eq. 6 the latest firing date of event e according to the parametric time interval associated to the transition $l(e)$. Also, an event e has a finite firing date iff it actually fires: this means that no other event e' in conflict with e can have a finite firing date e (Eq. 7). Finally with read arcs, in case the event e is weakly ordered before an event e', then with Eq. 8, e must fire before e'.

While Eqs. 5 to 7 define when an event can be fired, *i.e.* they give it a constrained but finite firing date, the last two equations define the cases in which an event cannot fire at all, giving it an infinite firing date. First, if one of the preconditions of event e has an infinite production date, then e has an infinite firing date (Eq. 9). Second, e may have an infinite firing date if it is in direct conflict with another event that has a finite firing date (Eq. 10). This implies that this event with a finite firing date will fire before e would have been forced to fire *i.e.* before its activity duration reaches the upper bound of the interval. Note that this is the only way to introduce infinite firing dates in the equation system. Those will then be propagated by Eq. 9.

Example 4. We consider the PSwPN of Fig. 1a. One of its TBP with symbolic constraints is presented on Fig. 3. For the values $a = 2$ and $b = 4$ of the parameters, a valid timing that verifies these constraints is $\theta(e_1) = \infty$, $\theta(e_2) = 3$, $\theta(e_3) = 8$, $\theta(e_4) = 15$ and $\theta(e_5) = \infty$.

3.2 Temporally Complete Time Branching Processes

Valid time branching processes as defined by Def. 10 and 11 do not necessarily contain correct executions, since a TBP is *a priori* incomplete in the sense that all timed constraints of the PSwPN may not be included yet in the TBP: by extending the TBP with additional events, new conflicts may appear that would add those constraints. We will therefore consider *temporally complete* TBP as defined below:

Definition 12 (Temporally complete TBP). *Let* $\mathcal{N} = \langle P, T, W, W_r, W_s, m_0, \mathsf{eft}, \mathsf{lft}, \Pi, D_\Pi \rangle$ *be PSwPN and* v *be a valuation of its parameters. A valid TBP* $\langle B, E, F, F_r, l, \theta \rangle$ *of* $\mathcal{N}_v$ *is* temporally complete *if for all the extensions* $\langle t, e \rangle$ *of* $\langle B, E, F, F_r, l \rangle$*,*

$$\forall e' \in E \ s.t. \ \theta(e') \neq \infty, \ \mathsf{adur}({}^*e, t, \theta(e')) \leq v(\mathsf{lft}(t)) \tag{11}$$

This definition basically says that the firing date of all events in the TBP should be less or equal than the latest firing date of all possible extensions. Since the conflicts that have not yet been discovered will result from these extensions, this implies that all the events in the TBP are possible before these conflicts occur. It further ensures that all the parallel branches in the TBP have been unfolded to a same date. A similar condition can be stated for time processes.

Example 5. For the TBP of Fig. 3, the timing given in example 4, although valid, admits the firing of t_2 as an extension after e_3, and its maximal firing date is 13 which is inferior to the firing date of e_4. Thus, this TPB cannot be complete.

3.3 Extensions of a TBP

We now show how a given TBP can be extended with additional events, eventually leading to the construction of the whole unfolding.

Proposition 1. *Let* $\mathcal{N}$ *be a PSwPN and* v *a valuation of its parameters. Let* $\langle B, E, F, F_r, l, \theta \rangle$ *be a temporally complete TBP of* $\mathcal{N}_v$ *and let* $\langle t, e \rangle$ *be an extension of* $\beta = \langle B, E, F, F_r, l \rangle$*. Let* β' *be the branching process obtained by extending* β *by* $\langle t, e \rangle$*. Then there exists* θ' *such that* $\langle \beta', \theta' \rangle$ *is a valid TBP of* $\mathcal{N}_v$*.*

While the TBP obtained by the extension $\langle t, e \rangle$ is valid, it is not necessarily temporally complete: only the conflicts present in β' are considered but e could be prevented by conflicts that have not yet been added through other extensions. We have the following result however:

Proposition 2. *Let* $\langle \beta, \theta \rangle$ *be a temporally complete TBP of a PSwPN and let* $\langle t, e \rangle$ *be the extension of* β *with the smallest latest firing date. Then* $\langle \beta, \theta \rangle$ *extended by* $\langle t, e \rangle$ *is a temporally complete TBP.*

3.4 Symbolic time branching processes

If we consider all the possible valuations of the parameters and all the possible valid timing functions for a given branching process of $\mathsf{Untimed}(\mathcal{N})$ we obtain what we call a *symbolic* TBP.

Definition 13 (Symbolic time branching process). *Let $\mathcal{N}$ be a PSwPN. A* symbolic time branching process *(STBP) Γ is a pair $\langle \beta, \mathcal{D} \rangle$ where $\beta = \langle B, E, F, F_r, l \rangle$ is a branching process of* $\mathsf{Untimed}(\mathcal{N})$*, $\mathcal{D}$ is a subset of $\mathbb{Q}^{|\Pi|} \times (\mathbb{R} \cup \{+\infty\})^{|E|}$ such that for all $\lambda = (v_1, \ldots, v_{|\Pi|}, \theta_1, \ldots, \theta_n, \ldots) \in \mathcal{D}$, if we note $E = \{e_1, \ldots, e_n, \ldots\}$, v_λ the valuation $(v_1, \ldots, v_{|\Pi|})$ and θ_λ the timing function such that $\forall i, \theta_\lambda(e_i) = \theta_i$, then $\langle \beta, \theta_\lambda \rangle$ is a valid TBP of $\mathcal{N}_{v_\lambda}$.*

In practice, the set $\mathcal{D}$ can be represented as a union of pairs $\langle \mathcal{E}_i, \mathcal{D}_i \rangle$ where $\mathcal{E}_i$ is a subset of the events of β and $\mathcal{D}_i$ is a rational convex polyhedron (possibly of infinite dimension) whose variables are the events in $\mathcal{E}_i$ plus the parameters of the net. Each point λ in $\mathcal{D}_i$ describes a value of the parameters and the finite values of the timing function on the elements of $\mathcal{E}_i$. For all elements not in $\mathcal{E}_i$, the timing function has value $+\infty$.

Now we can extend the notion of prefix to STBPs.

Definition 14 (Prefix of an STBP). *Let $\mathcal{N}$ be PSwPN whose set of parameters is Π. Let $\langle \beta, \bigcup_i \mathcal{E}_i, \bigcup_i \mathcal{D}_i \rangle$ and $\langle \beta', \bigcup_j \mathcal{E}'_j, \mathcal{D}' \rangle$ be two STBPs of $\mathcal{N}$. $\langle \beta, \mathcal{D} \rangle$ is a* prefix *of $\langle \beta', \mathcal{D}' \rangle$ if β is a prefix of β' and $\mathcal{D}$ is the projection of $\mathcal{D}'$ on the parameters plus the events of β.*

Finally, we can define the symbolic unfolding of a PSwPN.

Definition 15 (Symbolic unfolding). *The* symbolic unfolding *of a PSwPN $\mathcal{N}$ is the greatest STBP according to the prefix relation.*

This unfolding has the same size as the one computed for underlying Petri net. However, some events may not be able to take a finite firing date, in any circumstances. These events are not *possible* and will be useless. Thus, it will be sufficient to compute a prefix of the unfolding in which they are discarded.

3.5 Correctness and completeness

In this subsection we give two results proving the correctness and completeness of our symbolic unfolding w.r.t. to the concurrent semantics of (P)SwPNs, that we have given in Section 2 as time processes.

We first establish a result on the configurations of TBP. For every TBP $\Gamma = \langle B, E, F, F_r, l, v, \theta \rangle$, we define the set $E_{<\infty} = \{e \in E \mid \theta(e) < \infty\}$ of all the events which may fire in the TBP.

Proposition 3. *Let $\Gamma = \langle B, E, F, F_r, l, v, \theta \rangle$ be valid TBP. Then $E_{<\infty}$ is a configuration.*

The correctness result for our approach states that all the time processes we can extract from our TBPs, and in particular those contained in the symbolic unfolding, are valid:

Theorem 1 (Correctness). *Let $\mathcal{N} = \langle P, T, W, W_r, W_s, m_0, \mathsf{eft}, \mathsf{lft}, \Pi, D_\Pi \rangle$ be a parametric stopwatch Petri net and let $v \in D_\Pi$ be a valuation of its parameters. Let $\langle B, E, F, F_r, l, \theta \rangle$ be a temporally complete time branching process of $\mathcal{N}_v$. Let $E_{<\infty} = \{e \in E \mid \theta(e) < \infty\}$ and $\theta_{<\infty}$ is the restriction of θ to $E_{<\infty}$.*

$\langle E_{<\infty}, \theta_{<\infty} \rangle$ is a valid time process of $\mathcal{N}_v$.

Finally the following completeness result states that all valid time processes can be represented by a TBP. Therefore, since the symbolic unfolding contains all the valid TBPs, it also contains all the time processes of the PSwPN.

Theorem 2 (Completeness). *Let $\mathcal{N} = \langle P, T, W, W_r, W_s, m_0, \mathsf{eft}, \mathsf{lft}, \Pi, D_\Pi\rangle$ be a PSwPN and $v \in D_\Pi$ be a valuation of the parameters. Let $\langle B, E, F, F_r, l\rangle$ be a branching process of the underlying Petri net and $\langle E, \theta\rangle$ be a time process of the SwPN $\mathcal{N}_v$.*

There exists a temporally complete time branching process of $\mathcal{N}_v$, $\langle B', E', F', F'_r, l', \theta'\rangle$, such that $\forall e \in E, \exists e' \in E'$ s.t. $l(e) = l'(e')$ and $\theta(e) = \theta'(e')$.

The idea of the proof is to construct a TBP by adding all the events in conflict with some events of the time process.

4 Complete Prefixes of the Symbolic Unfolding

In this section, we show how to compute a complete prefix of the symbolic unfolding of a TPN. Consequently, from now we replace $v(\mathsf{eft}(t))$ by $\mathsf{eft}(t)$, $v(\mathsf{lft}(t))$ by $\mathsf{lft}(t)$, and we assume that $\mathsf{adur}(B, t, \theta) = \theta - \mathsf{TOE}(B, t)$, and ${}^{\circ}t = \emptyset$. In these conditions, we prove this prefix is finite.

A *consistent state* of the unfolding $\langle B, E, F, F_r, l, \mathcal{D}\rangle$ of a TPN $\mathcal{N} = \langle P, T, W, W_r, m_0, \mathsf{eft}, \mathsf{lft}\rangle$ is a pair $\langle A, \lambda\rangle$ such that $A \subseteq B$ is a cut and $\lambda \in \mathcal{D}$ and

- $\forall b \in A,\ \theta_\lambda({}^\bullet b) \neq \infty$,
- $\forall t \in T,\ {}^\bullet t \cup {}^{\circ}t \subseteq l(A) \Rightarrow \max_{b\in A}\{\theta_\lambda({}^\bullet b)\} \leq \mathsf{TOE}(t, A) + \mathsf{lft}(t)$.

To compute a finite prefix we need to consider a finite number of states. However, the firing dates of the events grow continuously in the unfolding. Therefore, we define an equivalence relation between two consistent states by considering the age of the tokens (a reduced age since even ages can grow infinitely). Finally, we prove that the same transitions are firable from two equivalent states.

Definition 16 (reduced age of a condition). *For any* co*-set A, any timing function θ, and any condition $b \in A$, we define the* (reduced) age *of b in A as*

$$\mathsf{age}(b, \theta, A) = \min\{\max_{b'\in A}\{\theta({}^\bullet b')\} - \theta({}^\bullet b), \max\{K(t) \mid t \in T \wedge t \in l(b)^\bullet\}\}$$

where $K(t) = \begin{cases} \mathsf{eft}(t) & \textit{if } \mathsf{lft}(t) = +\infty \\ \mathsf{lft}(t) & \textit{otherwise.} \end{cases}$

Definition 17 (Equivalent consistent states). *Two consistent states $\langle A_1, \lambda_1\rangle$ and $\langle A_2, \lambda_2\rangle$ are equivalent iff $l(A_1) = l(A_2)$ and $\forall b_1 \in A_1, \forall b_2 \in A_2,$ s.t. $l(b_1) = (b_2),\ \mathsf{age}(b_1, \theta_{\lambda_1}, A_1) = \mathsf{age}(b_2, \theta_{\lambda_2}, A_2)$.*

Theorem 3 (Firing a transition in equivalent states). *Let $s_1 = \langle A_1, \lambda_1\rangle$ and $s_2 = \langle A_2, \lambda_2\rangle$ be two equivalent consistent states of the unfolding $\langle B, E, F, F_r, l, \mathcal{D}\rangle$ of a TPN $\mathcal{N} = \langle P, T, W, W_r, m_0, \mathsf{eft}, \mathsf{lft}\rangle$. If a transition t is firable from s_1 in an event e_1 at a date $\theta_{\lambda_1}(e_1) \geq \max_{b\in A_1}(\theta_{\lambda_1}({}^\bullet b))$, before all the other enabled transitions (i.e. $\forall t \in \mathsf{en}(l(A_1))\theta_{\lambda_1}(e_1) \leq \mathsf{TOE}(t, A_1) + \mathsf{lft}(t)$), then*

1. *t is firable from s_2 in an event e_2 at the date $\theta_{\lambda_1}(e_1) - \max_{b \in A_1}(\theta_{\lambda_1}({}^\bullet b)) + \max_{b \in A_2}(\theta_{\lambda_2}({}^\bullet b))$, before all the other enabled transitions,*
2. *the states reached after the firing are equivalent.*

Knowing that the same behaviors are possible after equivalent states we can stop the construction of the unfolding by defining the notion of *cut-off* event.

Definition 18 (Cut-off event). *Let $\mathcal{N} = \langle P, T, W, W_r, m_0, \mathsf{eft}, \mathsf{lft}\rangle$ be a TPN. and let $\beta = \langle B, E, F, F_r, l, \mathcal{D}\rangle$ be a symbolic time branching process of $\mathcal{N}$. An event $e \in E$ is a* cut-off *event if there exists $e' \in E$ such that:*

- *$e' < e$,*
- *$l(e') = l(e)$,*
- *$\forall \lambda \in \mathcal{D},\ \exists \lambda' \in \mathcal{D}$ s.t. $\langle C_{e'}, \lambda'\rangle$ and $\langle C_e, \lambda\rangle$ are equivalent.*

Definition 19 (Cut-off-free maximal prefix). *Let $\mathcal{N}$ be a TPN and let $\Gamma = \langle \beta, D\rangle$ be its symbolic unfolding. The* cut-off-free maximal prefix *$CFP(\mathcal{N})$. is the greatest prefix of Γ that does not contain any cut-off events.*

We prove that the prefix computed contains at least the firing of each fireable transition of the unfolding, ad we show that this prefix is finite.

Theorem 4 (Completeness of the prefix). *Let $\mathcal{N} = \langle P, T, W, W_r, m_0, \mathsf{eft}, \mathsf{lft}\rangle$ be a TPN whose symbolic unfolding is $\langle B, E, F, F_r, l, \mathcal{D}\rangle$. Let $CFP(\mathcal{N}) = \langle B^*, E^*, F^*, F_r^*, l^*, \mathcal{D}^*\rangle$. Then $\forall \lambda \in \mathcal{D},\ \forall e \in E$ s.t. $\theta_\lambda(e) \neq \infty$, $\exists \lambda^* \in \mathcal{D}^*,\ \exists e^* \in E^*$, s.t. $\theta_{\lambda^*}(e^*) \neq \infty$ and $l(e^*) = l(e)$.*

Theorem 5 (Finiteness of the prefix). *For any (1-safe) time Petri net $\mathcal{N}$, the cut-off-free maximal prefix $CFP(\mathcal{N})$ is finite.*

5 Conclusion

In this paper we have proposed a new technique for the unfolding of safe parametric stopwatch Petri nets that allow a symbolic handling of both time and parameters. To the best of our knowledge, this is the first time that the parametric or stopwatch cases are addressed in the context of unfoldings. Moreover, when restricting to the subclass of safe time Petri nets, our technique compares well with the previous approach of [6]. It indeed provides a more compact unfolding, by eliminating the duplication of transitions, and also removes the need for read arcs in the unfolding. As a tradeoff, the constraints associated with the firing times of events may seem slightly more complex.

We have partly implemented the technique in our tool, Romeo, whose 2.9.0 version can currently compute unfoldings of safe time Petri nets. The computation of the finite prefix is however not yet implemented. We propose instead to couple the method with a supervision technique that makes the unfolding finite based on a finite set of observations. This approach, that also works with parameters and stopwaches, is detailled in [10] with a case study.

Further work includes investigating non-safe bounded models and application of the unfolding technique to revisit the problems of model-checking and control.

References

1. P. A. Abdulla, S. P. Iyer, and A. Nylen. Unfoldings of unbounded Petri nets. In *Proceedings of CAV*, volume 1855 of *LNCS*, pages 495–507. Springer, 2000.
2. T. Aura and J. Lilius. A causal semantics for time Petri nets. *Theoretical Computer Science*, 243(2):409–447, 2000.
3. P. Baldan, N. Busi, A. Corradini, and G. M. Pinna. Functorial concurrent semantics for Petri nets with read and inhibitor arcs. In *CONCUR*, volume 1877 of *LNCS*, pages 442–457. Springer, 2000.
4. B. Berthomieu and M. Diaz. Modeling and verification of time dependent systems using time Petri nets. *IEEE trans. on Soft. Eng.*, 17(3):259–273, 1991.
5. B. Berthomieu, D. Lime, O. H. Roux, and F. Vernadat. Reachability problems and abstract state spaces for time Petri nets with stopwatches. *Journal of Discrete Event Dynamic Systems - Theory and Applications (DEDS)*, 17(2):133–158, 2007.
6. T. Chatain and C. Jard. Complete finite prefixes of symbolic unfoldings of safe time Petri nets. In *Proceedings of ICATPN*, volume 4024 of *LNCS*, pages 125–145. Springer, 2006.
7. T. Chatain and C. Jard. Sémantique concurrente symbolique des réseaux de Petri saufs et dépliages finis des réseaux temporels. In *Proceedings of NOTERE*, Tozeur, Tunisia, May-June 2010. IEEE Computer Society Press.
8. J. Esparza. Model checking using net unfoldings. *Science of Computer Programming*, 23:151–195, 1994.
9. J. Esparza and K. Heljanko. *Unfoldings, A Partial-Order Approach to Model Checking.* Monographs in Theoretical Computer Science. Springer, 2008.
10. B. Grabiec, L.-M. Traonouez, C. Jard, D. Lime, and O. H. Roux. Diagnosis using unfoldings of parametric time Petri nets. In *Proceedings of FORMATS*, LNCS, Vienna, Austria, September 2010. Springer. To appear.
11. T.A. Henzinger, P.W. Kopke, A. Puri, and P. Varaiya. What's decidable about hybrid automata? *Journal of Computer and System Sciences*, 57:94–124, 1998.
12. V. Khomenko and M. Koutny. Branching processes of high-level Petri nets. In *Proceedings of TACAS*, volume 2619 of *LNCS*, pages 458–472. Springer, 2003.
13. Didier Lime and Olivier (H.) Roux. Formal verification of real-time systems with preemptive scheduling. *Journal of Real-Time Systems*, 41(2):118–151, 2009.
14. K. L. McMillan. Using unfolding to avoid the state space explosion problem in the verification of asynchronous circuits. In *Proceedings of CAV*, volume 663 of *LNCS*, pages 164–177. Springer, 1992.
15. P. M. Merlin. *A study of the recoverability of computing systems.* PhD thesis, Dep. of Information and Computer Science, University of California, Irvine, CA, 1974.
16. L.-M. Traonouez, D. Lime, and O. H. Roux. Parametric model-checking of stopwatch Petri nets. *Journal of Universal Computer Science (J.UCS)*, 15(17):3273–3304, December 2009.