Back in Time Petri Nets

Thomas Chatain, Claude Jard

Back in Time Petri Nets

Thomas Chatain[1] and Claude Jard[2]

[1] LSV, ENS Cachan, INRIA, CNRS, France
[2] LINA, Université de Nantes, France

Abstract. The time progress assumption is at the core of the semantics of real-time formalisms. It is also the major obstacle to the development of partial-order techniques for real-time distributed systems since the events are ordered both by causality and by their occurrence in time. Anyway, extended free choice safe time Petri nets (TPNs) were already identified as a class where partial order semantics behaves well. We show that, for this class, the time progress assumption can even be dropped (time may go back in case of concurrency), which establishes a nice relation between partial-order semantics and time progress assumption.

Key words: real-time models, time Petri nets, concurrency, partial-order semantics, extended free choice, time progress assumption

1 Introduction

Time Petri nets [10] are now a well established model to describe dynamic systems present in many areas. They are restricted to control aspects, but have the following interesting features: – it is a hybrid model mixing discrete state changes with the inclusion of continuous timed constraints; – it makes explicit the causal dependency relationships between state transitions; – it defines exclusive choices (conflicts) between transitions and by complementation, transitions that can be executed independently (in concurrency).

The standard semantics of time Petri nets is sequential and defines timed words formed by a succession of discrete transitions interspersed with timed transitions representing the passage of time. The consequence is that time progresses throughout the execution of the sequence.

However, in a sequence of transitions, it is possible that transitions are in concurrency (we say they are "concurrent"). The result in general (modulo the satisfaction of time constraints) is that the ordering of these transitions is arbitrary and that the reverse order is also possible. This observation led to the definition of a concurrent semantics for Petri nets in which the behaviors are defined by partial orders (processes in the jargon of Petri nets). This new semantics is mapped with the sequential semantics by considering the total orders compatible with the partial order (the "interleavings").

In this concurrent semantics, however, the time continues to evolve sequentially in a monotonic way. But regarding concurrent transitions, this can be

disputable. From the point of view of modeling, concurrent transitions can be executed in parallel, even on remote computers distributed over a communication network. In such an execution environment, it is not necessary to consider that an exact global clock is shared by the entire system. It is even realistic to consider that there is no shared clock in such a system and that time constraints must only be considered for transitions that are causally related. The original idea developed in this paper is to question the existence of a global clock. We keep of course the property that the execution of causally related transitions is totally ordered in time, but this constraint is released for execution of concurrent transitions. One way to do this is to ask **what would happen if we allowed the time to go back in the firing of a concurrent transition?**

We want that such "back in time" semantics preserves nevertheless the concurrent semantics. We show in this paper that it is possible in the case of extended free choice time Petri nets.

To do this, we begin by expressing the standard semantics by equipping tokens with their date of birth. In this new formulation, the constraint of time progression is made explicit. The elimination of this constraint defines a back in time semantics. We first show that it produces executions that are correct with respect to the standard semantics (Theorem 1). But the semantics does not exploit all the possibilities offered by the non-temporal sequencing of the execution of concurrent transitions. We thus develop the notion of completeness to suggest that the semantics must produce all linear extensions of timed processes. We propose a new back in time semantics, which is both correct and complete (Theorem 3).

Related work. There are several variants of semantics for time Petri Nets, according the fact that the time constraints are placed on transitions or places (see the survey [5]). They are not always comparable. An important point, however, is to take into account or not the notion of urgency. The absence of urgency (the "weak"-semantics [12]) substantially alters the theoretical power of the model as well as its ability to model temporal phenomena. All these variants did not have the audacity to question the linear flow of time, which could go back in time in some cases. The closest idea to be compared is that expressed by P. Niebert [11] in a model-checking method for timed automata. In his approach, time is seen as a ordinary symbolic variable that may regress in some transitions at the Time Transition System level associated with the model. The goal is to have a more compact representation of the state space.

Time and causality does not necessarily blend well in a non deterministic model such as Petri nets. The reason is that the mixture of conflict and urgency breaks the locality of the standard firing rule of transitions. This is the main difficulty that had to be overcome in recent years to finally define the notion of unfolding of time Petri nets [7]. It is therefore understandable why the subclass of extended free-choice Petri nets is often used when applied to timed and stochastic model [2] net. In this class, in fact, we just have to look at the considered transition and transitions in conflict with it to decide its firing: decision appears to be local, while in the general case one must consider all firable transitions

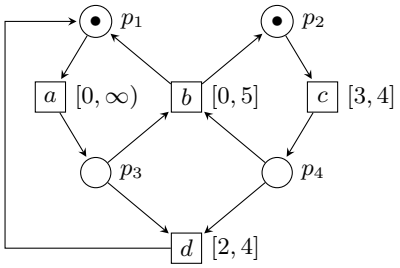


Fig. 1. An extended free choice safe time Petri net.

(those in conflict, but also those concurrent). It is therefore natural to use it when authorizing time to go back for concurrent transitions.

The rest of the paper is organized as follows. We first recall the standard semantics of time Petri nets with a special emphasis on the extended free-choice subclass. Then we propose an alternative semantics based on tokens, which we prove to be equivalent. In section 4, we present the partial order semantics (definition of processes). Section 5 presents two new (back in time) semantics. The first is proved to be correct (consistent with the processes). The second is proved to be correct and complete (defines all the linearizations of the processes). We end with a discussion and perspectives for the general case.

2 Time Petri Nets

2.1 Definition

Definition 1 (Time Petri Net (TPN) [10]). *A* time Petri net *is a tuple* $(P, T, pre, post, efd, lfd, M_0)$ *where* P *and* T *are finite sets of* places *and* transitions *respectively, pre and post map each transition* $t \in T$ *to its (nonempty)* preset *denoted* ${}^\bullet t \stackrel{\text{def}}{=} pre(t) \subseteq P$ *and its (possibly empty)* postset *denoted* $t^\bullet \stackrel{\text{def}}{=} post(t) \subseteq P$*; efd* $: T \longrightarrow Q$ *and lfd* $: T \longrightarrow Q \cup \{\infty\}$ *associate the* earliest firing delay $efd(t)$ *and* latest firing delay $lfd(t)$ *with each transition t;* $M_0 \subseteq P$ *is the* initial marking.

A time Petri net is represented as a graph with two types of nodes: places (circles) and transitions (rectangles). The interval $[efd(t), lfd(t)]$ is written near each transition (see Figure 1).

2.2 Standard Clocks-on-Transitions Semantics

State. A *state* of a time Petri net is given by a triple (M, ν, θ), where

- $M \subseteq P$ is a *marking* (ususally reprensented graphically by tokens in the places), from which we define the set of *enabled* transitions $En(M) \stackrel{\text{def}}{=} \{t \in T \mid {}^\bullet t \in M\}$;
- $\nu : En(M) \to \mathbb{R}$ assigns a clock value to every enabled transition;

– $\theta \in \mathbb{R}$ is the current time. It is not required for defining the semantics, but we use it here for convenience and to ease the comparison with the alternative semantics that we give later.

A time Petri net starts in an *initial state* $(M_0, \nu_0, 0)$, which is given by the *initial marking* M_0. Initially, all the clocks are reset: for every $t \in En(M_0)$, $\nu_0(t) \stackrel{\text{def}}{=} 0$. From state (M, ν, θ), two types of actions are possible:

Time delay. The TPN can wait until time θ' and reach state (M, ν', θ') with $\nu'(t) \stackrel{\text{def}}{=} \nu(t) + (\theta' - \theta)$ for every $t \in En(M)$, provided

– time progresses: $\theta' \geq \theta$; and
– no enabled transition overtakes its maximum delay:
$\forall t \in En(M) \quad \nu'(t) \leq lfd(t)$.

The time delay is written $(M, \nu, \theta) \xrightarrow{\theta'} (M, \nu', \theta')$.

Discrete action. Transition t can fire from state (M, ν, θ), if:

– t is enabled: $t \in En(M)$;
– t has reached its minimum firing delay: $\nu(t) \geq efd(t)$;

Firing transition t from state (M, ν, θ) leads to state (M', ν', θ), with $M' \stackrel{\text{def}}{=} (M \setminus {}^\bullet t) \cup t^\bullet$ and $\nu'(t) \stackrel{\text{def}}{=} \nu(t)$ if $t \in En(M \setminus {}^\bullet t)$ and $\nu'(t) \stackrel{\text{def}}{=} 0$ if $t \in En(M') \setminus En(M \setminus {}^\bullet t)$ (t is called "newly enabled"). We write $(M, \nu, \theta) \xrightarrow{t} (M', \nu', \theta)$.

Timed words. When representing an execution, we often forget the information about the intermediate states and delays, and remember only the sequence $((t_1, \theta_1), \ldots, (t_n, \theta_n))$ of transitions with their firing dates. This representation is called a *timed word*. The empty timed word is denoted ϵ.

2.3 Assumptions

Boundedness. All the TPNs we consider in this article are *1-bounded*, or *safe*, i.e. in every reachable state (M, ν, θ), if a transition t can fire from (M, ν, θ), then $t^\bullet \cap (M \setminus {}^\bullet t) = \emptyset$. Hence there is never more than one token in a place. This assumption is generally done in TPNs. Considering several tokens in places requires to define complex service policies [4].

Moreover, for technical simplicity, we require that even the untimed support is safe, i.e. the TPN remains safe if one replaces all the earliest firing delays by 0 and all the latest firing delays by ∞.

Time divergence. Finally we assume that time *diverges*: in every infinite timed word accepted by the TPN, time diverges to infinity. This assumption is very usual and comes from the intuition that no physical system can execute infinitely many action in finite time.

2.4 Extended Free Choice Time Petri Nets

In this article we will focus on a common class of TPNs which have many interesting properties, among which some concerning partial-order semantics and time.

Two transitions t and t' are in *conflict* if they share an input place. An extended free choice Petri net is a Petri net where every two transitions in conflict, have exactly the same input places.

Definition 2 (extended free choice [3, 8]). *An* extended free choice *(time) Petri net is a (time) Petri net such that:*

$$\forall t, t' \in T \quad {}^\bullet t \cap {}^\bullet t' \neq \emptyset \implies {}^\bullet t = {}^\bullet t' \ .$$

It is technically convenient to focus on normalized extended free choice TPNs, where every two transitions in conflict have the same latest firing delay.

Definition 3 (Normalization of an Extended Free Choice TPN). *Given an extended free choice TPN $N = (P, T, pre, post, efd, lfd, M_0)$, for every transition t we define $lfd'(t) \stackrel{\text{def}}{=} \min(\{lfd(t') \mid t' \in T \wedge {}^\bullet t \cap {}^\bullet t' \neq \emptyset\})$. Then we define the set $T' \subseteq T$ of transitions satisfying $efd(t) \leq lfd'(t)$. The normalization of N is the TPN $N' \stackrel{\text{def}}{=} (P, T', pre_{|T'}, post_{|T'}, efd_{|T'}, lfd'_{|T'}, M_0)$.*

For the TPN of Figure 1, normalization lowers $lfd(b)$ from 5 to 4 because transition d which is in conflict with b, has a smaller latest firing delay than b.

Lemma 1. *The semantics of extended free choice safe TPNs is unchanged by normalization: the original TPN and its normalized have the same set of states, and the possible delays and discrete actions are the same (they induce exactly the same timed transition system).*

Proof. Replacing $lfd(t)$ by $lfd'(t)$ does not change the behaviour: in an extended free choice TPN, the transitions in conflict with t are enabled at the same time as t. So one of them has to fire before the minimum of their latest firing delays expires. This firing disables the others.

Finally it is obvious that transitions having $efd(t) > lfd'(t)$ can never fire. □

3 Clocks-on-Tokens Semantics

In the previous section we have defined the semantics of TPNs in the usual way. But our work is based on another way to express it, where the clocks are assigned to the tokens instead of the enabled transitions. This semantics is equivalent to the previous one (at least in the case of safe Petri nets) in the sense that both semantics are timed bisimilar.

State. A *state* of a time Petri net is now given by a triple (M, dob, θ), where $M \subseteq P$ is the *marking*, $\theta \in \mathbb{R}$ is the current time and $dob : M \longrightarrow \mathbb{R}$ associates

a *date of birth* $dob(p) \in \mathbb{R}$ with each token (marked place) $p \in M$.

The marking and the current time play exactly the same role as in the clocks-on-transitions semantics, and the set of enabled transitions is defined the same way. But for every transition t enabled in a state (M, dob, θ), we define its *date of enabling* $doe(t)$ as the date of birth of the youngest token in its input places:

$$doe(t) \stackrel{\text{def}}{=} \max_{p \in {}^\bullet t} dob(p)\,.$$

The *initial state* is now $(M_0, dob_0, 0)$ and initially, all the tokens carry the date 0 as date of birth: for all $p \in M_0$, $dob_0(p) \stackrel{\text{def}}{=} 0$.

Time delay. The TPN can wait until time θ' provided

- time progresses: $\theta' \geq \theta$;
- no enabled transition overtakes its maximum delay:
  $\forall t \in En(M) \quad \theta' \leq doe(t) + lfd(t)$.

The reached state is simply (M, dob, θ'), and we write $(M, dob, \theta) \xrightarrow{\theta'} (M, dob, \theta')$.

Discrete action. Transition t can fire from state (M, dob, θ) if:

- t is enabled: $t \in En(M)$;
- t has reached its minimum firing delay: $\theta \geq doe(t) + efd(t)$;

Firing transition t from state (M, dob, θ) leads to state (M', dob', θ), with $M' \stackrel{\text{def}}{=} (M \setminus {}^\bullet t) \cup t^\bullet$ and $dob'(p) \stackrel{\text{def}}{=} dob(p)$ if $p \in M \setminus {}^\bullet t$ and $dob'(p) \stackrel{\text{def}}{=} \theta'$ if $p \in t^\bullet$ (by assumption the two cases are exclusive). We write $(M, dob, \theta) \xrightarrow{t} (M', dob', \theta)$.

3.1 Timed Bisimulation between the two Semantics

Definition 4 (Timed Transition System). *A* timed transition system *(TTS) is a tuple $(S, s_0, \Sigma, \to)$ where S is a set of states, $s_0 \in S$ is the initial state, Σ is a finite set of actions disjoint from $\mathbb{R}_{\geq 0}$ (for the case of the semantics of a TPN, Σ is the set T of transitions), and $\to \subseteq S \times (\Sigma \cup \mathbb{R}_{\geq 0}) \times S$ is a set of edges. We write $s \xrightarrow{e} s'$ for $(s, e, s') \in \to$.*

Definition 5 (Timed bisimulation). *Let $T_1 = (S_1, s_1^0, \Sigma, \to_1)$ and $T_2 = (S_2, s_2^0, \Sigma, \to_2)$ be two TTS. We say that T_1 and T_2 are* timed bisimilar *if there exists a binary relation $\approx$ between S_1 and S_2, called* timed bisimulation relation*, such that:*

- $s_1^0 \approx s_2^0$,
- *if $s_1 \xrightarrow{a}_1 s_1'$ with $a \in \Sigma \cup \mathbb{R}_{\geq 0}$ and $s_1 \approx s_2$, then $\exists s_2'$ such that $s_2 \xrightarrow{a}_2 s_2'$ and $s_1' \approx s_2'$; conversely if $s_2 \xrightarrow{a}_2 s_2'$ with $a \in \Sigma \cup \mathbb{R}_{\geq 0}$ and $s_1 \approx s_2$, then $\exists s_1'$ such that $s_1 \xrightarrow{a}_1 s_1'$ and $s_1' \approx s_2'$.*

Lemma 2. *For every safe TPN, the TTS induced by the clocks-on-tokens semantics is timed bisimilar to the TTS induced by the clocks-on-transitions semantics.*

Proof. Every state $S = (M, dob, \theta)$ of the clocks-on-tokens semantics maps to a state $f(S) \stackrel{\text{def}}{=} (M, \nu, \theta)$ with $\nu(t) \stackrel{\text{def}}{=} \theta - doe(t)$. Note that f may not be injective. Then we define the relation $\approx$ as:

$$(M, \nu, \theta) \approx (M, dob, \theta) \iff (M, \nu, \theta) = f((M, dob, \theta))$$

and we show that $\approx$ is a timed bisimulation relation. The relation between initial states is immediate. Consider now related states $(M, \nu, \theta) \approx (M, dob, \theta)$.

- $(M, \nu, \theta) \xrightarrow{\theta'} (M, \nu', \theta')$
  iff $\forall t \in En(M) \quad \nu(t) + (\theta' - \theta) \leq lfd(t)$
  iff $\forall t \in En(M) \quad \theta' - doe(t) \leq lfd(t)$
  iff $(M, dob, \theta) \xrightarrow{\theta'} (M, dob, \theta')$
- If $(M, dob, \theta) \xrightarrow{t} (M', dob', \theta)$, then $(M, \nu, \theta) \xrightarrow{t} f((M', dob', \theta))$ (this is immediate by the definition of f).
- If $(M, \nu, \theta) \xrightarrow{t} (M', \nu', \theta)$, then $(M, dob, \theta) \xrightarrow{t} (M', dob', \theta)$ with $dob'(p) = dob(p)$ for every $p \in M \setminus {}^\bullet t$ and $dob'(p) = \theta$ for every $p \in t^\bullet$. We only need to check that $f((M', dob', \theta)) = (M', \nu', \theta)$. □

4 Partial Order Representation of Runs: Processes

We will define the mapping Π from the timed words of a safe time Petri net to their partial order representation as processes. These processes are those described in [1]. We use a canonical coding like in [9].

Coding of Events and Conditions. Each process will be a set $\mathcal{E}$ of pairs $(e, \mathcal{E}(e))$, where e is an *event* and $\mathcal{E}(e) \in \mathbb{R}$ is its firing date. We denote $E_\mathcal{E}$ (or simply E) the set of events in $\mathcal{E}$. Each event e is itself a pair $({}^\bullet e, \tau(e))$ that codes an occurrence of the transition $\tau(e)$ in the process. ${}^\bullet e$ is a set of pairs $b \stackrel{\text{def}}{=} ({}^\bullet b, place(b))$. Such a pair is called a *condition* and refers to the token that has been created by the event ${}^\bullet b$ in the place $place(b)$. We say that the event $e \stackrel{\text{def}}{=} ({}^\bullet e, \tau(e))$ *consumes* the conditions in ${}^\bullet e$. Symmetrically the set $\{(e, p) \mid p \in \tau(e)^\bullet\}$ of conditions that are *created* by e is denoted $e^\bullet$. A virtual initial event $\perp$ is used as ${}^\bullet b$ for initial conditions. By convention $\perp^\bullet \stackrel{\text{def}}{=} \{\perp\} \times M_0$ and $\mathcal{E}(\perp) \stackrel{\text{def}}{=} 0$. We write $E_\perp$ for $E \cup \{\perp\}$.

For all set B of conditions, we denote $Place(B) \stackrel{\text{def}}{=} \{place(b) \mid b \in B\}$, and when the restriction of $place$ to B is injective, we denote $place^{-1}_{|B}$ its inverse, and for all $P \subseteq Place(B)$, $Place^{-1}_{|B}(P) \stackrel{\text{def}}{=} \{place^{-1}_{|B}(p) \mid p \in P\}$. The set of conditions that remain at the end of the process $\mathcal{E}$ (meaning that they have

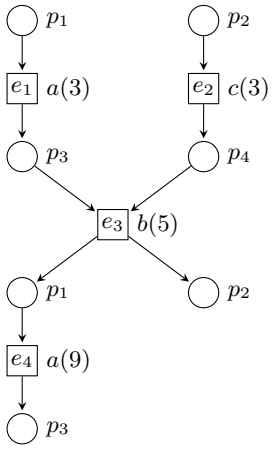


Fig. 2. A representation of a process of the time Petri net of Figure 1. The dates of the events are in brackets.

been created by an event of E, and no event of E has consumed them) is $\uparrow(E) \stackrel{\text{def}}{=} \bigcup_{e \in E_\perp} e^\bullet \setminus \bigcup_{e \in E} {}^\bullet e$.

To summarize the coding of the processes, it is convenient to define a set D_N, such that all the events that appear in the processes of a Petri net N, are elements of D_N.

Definition 6 (D_N). *We define D_N as the smallest set such that*

$$\forall B \subseteq \bigcup_{e \in D_{N\perp}} e^\bullet \quad \forall t \in T \quad Place(B) = {}^\bullet t \implies (B, t) \in D_N \ .$$

Notice that this inductive definition is initialized by the fact that $\perp \in D_{N\perp}$.

Causality. We define the relation $\rightarrow$ on the events as: $e \rightarrow e' \iff e^\bullet \cap {}^\bullet e' \neq \emptyset$. The reflexive transitive closure $\rightarrow^*$ of $\rightarrow$ is called the *causality* relation. Two events of a process that are not causally related are called *concurrent*. For all event e, we denote $\lceil e \rceil \stackrel{\text{def}}{=} \{f \in E \mid f \rightarrow^* e\}$, and for all set E of events, $\lceil E \rceil \stackrel{\text{def}}{=} \bigcup_{e \in E} \lceil e \rceil$.

Dates. We define the date of birth of the token which stands in a place $p \in Place(\uparrow(E))$ after a process $\mathcal{E}$ as $dob_{\mathcal{E}}(p) \stackrel{\text{def}}{=} \mathcal{E}({}^\bullet(place^{-1}_{|\uparrow(E)}(p)))$.

This allows us to define the state that is reached after a process $\mathcal{E}$ of N as: $RS(\mathcal{E}) \stackrel{\text{def}}{=} (Place(\uparrow(E)), dob_{\mathcal{E}}, \max_{e \in E_\perp} \mathcal{E}(e))$. Actually we will use this definition even for sets $\mathcal{E}$ of dated events which are not processes.

Definition 7. *The function Π that maps each timed word $((t_1, \theta_1), \ldots, (t_n, \theta_n))$ to a process is defined as follows:*

– $\Pi(\epsilon) \stackrel{\text{def}}{=} \emptyset$
– $\Pi(((t_1,\theta_1),\ldots,(t_{n+1},\theta_{n+1}))) \stackrel{\text{def}}{=} \mathcal{E} \cup \{(e,\theta_{n+1})\}$, *where*
  • $\mathcal{E} \stackrel{\text{def}}{=} \Pi(((t_1,\theta_1),\ldots,(t_n,\theta_n)))$ *and*
  • *the event* $e \stackrel{\text{def}}{=} (Place^{-1}_{|\uparrow(E)}({}^\bullet t_{n+1}), t_{n+1})$ *represents the last firing of the sequence.*

In the representation of a process, the rectangles represent the events, and the circles represent the conditions. An arrow from a condition b to an event e means that $b \in {}^\bullet e$. An arrow from an event e to a condition b means that $e = {}^\bullet b$.

Figure 2 shows a process of the TPN of Figure 1, which is the image by Π of the two timed words $((a,3),(c,3),(b,5),(a,9))$ and $((c,3),(a,3),(b,5),(a,9))$: permuting the concurrent events corresponding to the first occurrences of a and c yields two different timed words for the same process. Yet these permutations are very limited because the sequential semantics forces time to progress. We will obtain much more permutations in Section 5 with our relaxed semantics.

4.1 Characterization of Processes

Since timed processes are defined as sets of dated events (the events being taken from the set D_N defined above), a natural problem is to decide whether a subset of dated events is a process. In the general case, the answer is nontrivial and was treated in [1]. But the case of extended free choice time Petri nets behaves much better and we recall here how to characterize processes in this case. We assume that the net is normalized, which simplifies even more the formulation.

Lemma 3 (Characterization of processes). *Let N be a normalized safe extended free choice TPN and let $\mathcal{E} \subseteq D_N \times \mathbb{R}$ be a finite set of dated events (we denote E the set of events in $\mathcal{E}$). $\mathcal{E}$ is a process of N iff:*

– *E is causally closed: $\lceil E \rceil = E$;*
– *E is conflict free: $\nexists e, e' \in E \quad e \neq e' \ \wedge \ {}^\bullet e \cap {}^\bullet e' \neq \emptyset$;*
– *firing delays are met: $\forall e \in E \quad efd(\tau(e)) \leq \mathcal{E}(e) - doe(e) \leq lfd(\tau(e))$;*
– *$\mathcal{E}$ is* temporally complete*: every transition t enabled in the state $RS(\mathcal{E})$ reached after $\mathcal{E}$, satisfies $doe(t) + lfd(t) \leq \max_{e \in E_\perp} \mathcal{E}(e)$.*

Example. Consider the process of Figure 2 and replace the firing dates by parameters $\mathcal{E}(e_1)$, $\mathcal{E}(e_2)$, $\mathcal{E}(e_3)$, $\mathcal{E}(e_4)$. The values of the parameters that correspond to processes accepted by the TPN of Figure 1 are those satisfying the following constraints, given by earliest and latest firing delays of the events

$$\begin{cases} 0 \leq \mathcal{E}(e_1) \leq \infty \\ 3 \leq \mathcal{E}(e_2) \leq 4 \\ 0 \leq \mathcal{E}(e_3) - \max\{\mathcal{E}(e_1), \mathcal{E}(e_2)\} \leq 4 \qquad \text{(after normalization } lfd(b) \text{ is set to 4)} \\ 0 \leq \mathcal{E}(e_4) - \mathcal{E}(e_3) \leq \infty \end{cases}$$

plus the constraint that the process is temporally complete:

$$\max\{\mathcal{E}(e_1), \mathcal{E}(e_2), \mathcal{E}(e_3), \mathcal{E}(e_4)\} - \mathcal{E}(e_3) \leq 4$$

i.e., here, that transition c, which is enabled in the final state, has not fired yet at the time of the end of the process.

It is worth noticing that every set $\mathcal{E}$ satisfying all the conditions but the last, is a prefix (i.e. a causally closed subset) of a process. The following lemma is a corollary of Theorem 3.2 of [6] and a strong property of extended free choice TPNs. The fact that it does not hold for general safe TPNs is at the origin of the difficulties to define their unfoldings.

Lemma 4 (Characterization of prefixes of processes). *Let N be a normalized safe extended free choice TPN satisfying the diverging time assumption and let $\mathcal{E} \subseteq D_N \times \mathbb{R}$ be a finite set of dated events (we denote E the set of events in $\mathcal{E}$). $\mathcal{E}$ is a prefix of a process of N iff:*

- *E is causally closed: $\lceil E \rceil = E$;*
- *E is conflict free: $\nexists e, e' \in E \quad e \neq e' \ \wedge \ {}^\bullet e \cap {}^\bullet e' \neq \emptyset$;*
- *firing delays are respected: $\forall e \in E \quad efd(\tau(e)) \leq \mathcal{E}(e) - doe(e) \leq lfd(\tau(e))$.*

Proof. By construction (and by Lemma 3), all the prefixes of the processes satisfy the constraints. We show the converse by induction on the number of events in $\mathcal{E}$. The case of the empty process is immediate.

Consider now a nonempty process $\mathcal{E}$ satisfying the constraints. Choose an event e of $\mathcal{E}$ with maximal date; if there are several candidates, choose one which is also maximal w.r.t. causality (causality is acyclic by construction of D_N). The selected event e is maximal w.r.t. causality among all the events of $\mathcal{E}$: the inequality about the earliest firing delays, with the fact that earliest firing delays are non negative, prevent events with strictly smaller date than e from being causal successors of e.

Process $\mathcal{E}$ without event e is causally closed, conflict free and satisfies the inequalities about the firing delays. Assume, by the induction hypothesis, that it is a prefix of a process $\mathcal{F}$. This process can be truncated if necessary so that all the events that are in $\mathcal{F}$ and not in $\mathcal{E}$ have a date strictly smaller than the date $\mathcal{E}(e)$ of e. This means that the current date in the state $RS(\mathcal{F})$ reached after $\mathcal{F}$, is smaller or equal to $\mathcal{E}(e)$. Therefore, if $\mathcal{F}$ contains an event f in conflict with e, then f can be dropped (together with its causal successors, if any): this still yields an acceptable $\mathcal{F}$ because, N being extended free choice and normalized, e and f have exactly the same input conditions, the same date of enabling and the same latest firing delay and this delay is not expired at time $\mathcal{E}(e)$.

Hence we can assume that $\mathcal{F}$ has no event in conflict with e. The idea is to add e to $\mathcal{F}$. But, because of their latest firing delay, some transitions enabled in $RS(\mathcal{F})$ may not be allowed to wait until time $\mathcal{E}(e)$. We can simply fire them and complete process $\mathcal{F}$ with the corresponding events, as long as such transitions remain. The diverging time assumption ensures that this terminates; and, for the same reason as before, no event in conflict with e needs to be added (they can wait until $\mathcal{E}(e)$). Finally e can be added, and $\mathcal{E}$ is a prefix of the process that we obtain. □

5 Back in Time Semantics

We now explore the semantics of time Petri nets when the time progress assumption is dropped.

Of course dropping the time progress assumption will generate new timed words. But we will show that for free choice TPNs, the relaxed clocks-on-tokens semantics remains related to the classical semantics in the sense that it produces the same partial-order executions.

5.1 Relaxed clocks-on-tokens semantics

Definition 8 (relaxed clocks-on-tokens semantics). *The relaxed semantics is obtained simply by dropping the constraint of time progress $\theta' \geq \theta$ in the time delays. The constraint that no enabled transition overtakes its maximum delay, remains: $\forall t \in En(M) \quad \theta' \leq doe(t) + lfd(t)$.*

Hence time may go back in the past without any restriction. On the other hand, the constraint about maximum delays prevents time from progressing too much.

Timed words and processes. We define timed words like for the classical semantics, as sequences $((t_1, \theta_1), \ldots, (t_n, \theta_n))$ of transitions with their firing dates. Simply, now, the ordering $\theta_1 \leq \cdots \leq \theta_n$ may not hold. This does not prevent us from defining processes from these timed words, using the same mapping Π as for the classical semantics. We show that the processes that we obtain are prefixes of processes of the classical semantics.

Example. As an example, consider the TPN of Figure 1. The timed word $(a, 4), (c, 3)$ is accepted by the relaxed clocks-on-tokens semantics: after firing a at time 4, p_2 and p_3 are marked, with $dob(p_2) = 0$ and $dob(p_3) = 4$, and the current time is 4. After that, time may go back to 3, and c can fire.

Theorem 1 (Correctness). *Let N be an extended free choice TPN. Every process of N under the relaxed clocks-on-tokens semantics, is a prefix of a process of N under the classical semantics.*

Proof. We show, by induction on the length, that for every timed word w accepted by N under the relaxed clocks-on-tokens semantics, all the events of $\Pi(w)$ satisfy the conditions of Lemma 4 and that the state $RS(w)$ reached after $\Pi(w)$ is the state reached after w. The case of the empty timed word is immediate.

Now, assume that transition t fires at time θ from the state reached after a timed word w that satisfies the induction hypothesis. Let e be the event corresponding this firing of $t = \tau(e)$ at time $\theta = \mathcal{E}(e)$. The constraint that no enabled transition overtakes its maximum delay during the delay to time θ, is satisfied in particular by θ, and guarantees that $\mathcal{E}(e) - doe(e) \leq lfd(t)$. And the constraint about the earliest firing delay of t is checked when t fires, which implies that $\mathcal{E}(e) - doe(e) \geq efd(t)$. □

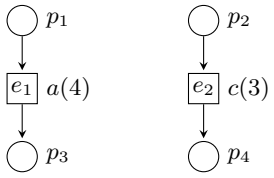


Fig. 3. A process of the time Petri net of Figure 1. The dates of the events are in brackets. This process has two linearizations: $((c,3),(a,4))$ and $((a,4),(c,3))$. Only the first one is accepted by the classical semantics, but both are accepted by the relaxed clocks-on-tokens semantics.

Definition 9 (Linearization). *Let $\mathcal{E}$ be a process or a prefix of a process and let L be a bijection from E to $\{1,\ldots,|E|\}$ such that for every i,j, $L(i) \to L(j) \implies i < j$ (i.e. L respects the causal ordering). Then the timed word $((\tau(L(1)),\mathcal{E}(L(1))),\ldots,(\tau(L(n)),\mathcal{E}(L(n))))$ is called a* linearization *of $\mathcal{E}$.*

Corollary 1. *Every timed word accepted by N under the relaxed clocks-on-tokens semantics, is a prefix of a permutation of a timed word accepted by N under the classical clocks-on-tokens semantics.*

Proof. By construction of the processes, every timed word w accepted by N under the relaxed clocks-on-tokens semantics, is a linearization of the process $\Pi(w)$, which is also a process of N under the classical semantics. □

Example. As an example, consider again the TPN N of Figure 1. The timed word $((a,4),(c,3))$, which is accepted by the relaxed clocks-on-tokens semantics, is a linearization of $\Pi((c,3),(a,4))$, represented on Figure 3, which is a process of N under the classical semantics.

5.2 More relaxed clocks-on-tokens semantics

Our previous subsection was based on a minimalist modification of the classical semantics: we have simply dropped the time progress assumption. We have shown that this preserves the partial order behaviour and that the timed words accepted under the relaxed semantics are linearizations of processes under the classical semantics.

Yet not all linearizations of processes are accepted by the relaxed clocks-on-tokens semantics. For example, for the TPN N of Figure 1, $(a,5),(c,3)$ is a linearization of a process, but it is not accepted by the relaxed semantics. The reason is that $lfd(c) < 5$, i.e. the latest firing delay of c prevents time from progressing up to 5 while c is enabled. Still a and c are concurrent, which suggests that, if one makes abstraction of the temporal ordering, they can fire independently in any order: in particular one could fire a and then – after possibly going back in time – fire c.

For this we need to relax a bit more the constraints about time progress, and let time go back, but also go forth without any restriction as long as discrete

actions are not concerned. Hence, only when firing a transition t, one checks that the earliest and latest firing delays are respected, and this check can be done only locally (on the transitions in conflict with t) since concurrent transitions are not concerned.

Definition 10 (more relaxed clocks-on-tokens semantics). *The more relaxed clocks-on-tokens semantics is defined from the clocks-on-tokens semantics, with the following changes.*

Time delay. *All the constraints on θ' for the time delay are dropped: the TPN can now reach any time θ' from any state.*

Discrete action. *But transition t can fire from state (M, dob, θ) if:*

- *t is enabled: $t \in En(M)$;*
- *t has reached its minimum firing delay: $\theta \geq doe(t) + efd(t)$;*
- *no transition in conflict with t overtakes its maximum firing delay: $\forall t' \in T \quad {}^\bullet t' \cap {}^\bullet t \neq \emptyset \implies \theta \leq doe(t') + lfd(t')$.*

Notice that after normalization of an extended free choice TPN, the constraint about latest firing delays can simply be checked on the transition t itself: $\theta \leq doe(t) + lfd(t)$.

Example. Consider again the TPN of Figure 1. It can now wait until time 5 without firing any transition, fire a, and then go back to time 3 and fire c.

Again correctness of the more relaxed clocks-on-tokens semantics is expressed in terms of preservation of processes.

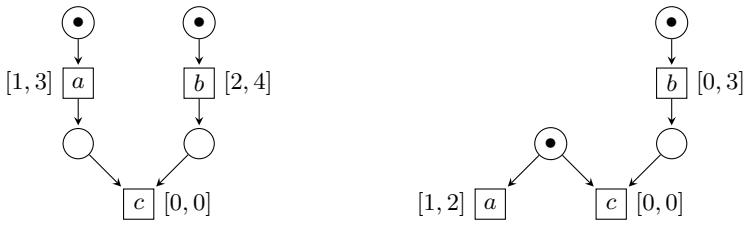
Theorem 2 (Correctness). *Let N be an extended free choice TPN. Every process of N under the more relaxed clocks-on-tokens semantics, is a prefix of a process of N under the classical clocks-on-tokens semantics.*

Proof. Like for Theorem 1, we show, by induction on the length, that for every timed word w accepted by N under the more relaxed clocks-on-tokens semantics, all the events of $\Pi(w)$ satisfy the conditions of Lemma 4 and that the state $RS(w)$ reached after $\Pi(w)$ is the state reached after w. The difference is that, with the more relaxed semantics, the condition about the latest firing delay is not checked during the delay to θ, but when firing t. □

Theorem 3 (Completeness). *The timed words accepted by the more relaxed clocks-on-tokens semantics of an extended free choice TPN coincide precisely with the linearizations of the prefixes of its processes under the classical semantics.*

Proof. That the more relaxed clocks-on-tokens semantics generates only linearizations of prefixes of its processes under the classical semantics, is a direct corollary of Theorem 2.

It remains to check that it generates all of them. We proceed by induction on the number of events in the prefix. The case of the empty prefix is immediate. Let $\mathcal{E}$ be a nonempty process under the classical semantics, and



(a) Counter-example for the clocks-on-transitions semantics

(b) Counter-example for non free choice TPNs

Fig. 4. Counter-examples

$((t_1, \theta_1), \ldots, (t_n, \theta_n))$ be a linearization of $\mathcal{E}$. By definition of the linearization, $((t_1, \theta_1), \ldots, (t_{n-1}, \theta_{n-1}))$ is also a linearization of a prefix. Assume, by the induction hypothesis, that it is accepted under the more relaxed semantics. It remains to show that the more relaxed semantics lets transition t_n fire at time θ_n from the state reached after $((t_1, \theta_1), \ldots, (t_{n-1}, \theta_{n-1}))$. The constraint of Lemma 4 applied to $\mathcal{E}$ gives precisely this constraint for the last event of the linearization (the one corresponding to the firing of t_n at time θ_n). □

6 Discussion and Perspectives

Relaxed clocks-on-transitions semantics. As an alternative to our relaxed clocks-on-transitions semantics, one could envisage starting from the classical clocks-on-transitions semantics and dropping the constraint $\theta \leq \theta'$.

But even for free choice TPNs, timed words accepted by the relaxed clocks-on-transitions semantics are not necessarily permutations of timed words accepted by the classical clocks-on-transitions semantics. For the TPN of Figure 4(a), the relaxed clocks-on-transitions semantics accepts the timed word $((a, 3), (b, 2), (c, 2))$: it allows one to fire a at time 3, followed by b at time 2. In the reached state, c is enabled and its clock has value $\nu(c) = 0$ since it was enabled by the firing of b. As a result, c fires at time 2. In the classical clocks-on-transitions semantics, firing b at time 2 and a at time 3 implies firing c at time 3.

This shows that the clocks-on-tokens semantics, although equivalent to the clocks-on-transitions semantics in the classical setting, is more robust to variations of the semantics. Another argument in this sense is given in [4] for the case of unbounded time Petri nets.

Counter-example for non free choice TPNs. The TPN of Figure 4(b) is not free choice since transitions a and b are in conflict, but do not have the same presets. With any of the relaxed semantics that we defined, a can fire at time 1 from the initial state, and then b can fire at time 0. This is not a linearization of any process because in the classical semantics, firing b at time 0 implies firing c at time 0 and thus prevents a from firing.

Perspectives. We believe that our result can be adapted quite easily to other formalisms for real-time concurrent systems (arc-timed Petri nets, networks of timed automata...) provided a restriction on local choices is done (corresponding to the extended free choice assumption for TPNs). Without this assumption, the previous counter-example already shows how transitions that are apparently concurrent can influence one the other. Then a back in time semantics for the general case should be constrained by these dependencies. Some of these dependencies played a role in the definition of unfoldings for TPNs, but identifying them precisely remains a challenge.

Finally, the algorithms for the analysis of timed models construct and solve systems of linear constraints on temporal parameters (like occurrence time, value of clock...) By relaxing the constraints about time progress, our back in time semantics would generate fewer inequalities and fewer symbolic states. And it preserves properties like fireability of a transition or reachability of a place. We plan to experiment the construction of the symbolic state graph generated by our semantics and evaluate how much it improves the analysis algorithms.

References

1. T. Aura and J. Lilius. A causal semantics for time Petri nets. *Theoretical Computer Science*, 243(1-2):409–447, 2000.
2. A. Benveniste, E. Fabre, and S. Haar. Markov nets: probabilistic models for distributed and concurrent systems. *IEEE Trans. Automat. Contr.*, 48(11):1936–1950, 2003.
3. E. Best. Structure theory of Petri nets: the free choice hiatus. In *Proceedings of an Advanced Course on Petri Nets: Central Models and Their Properties, Advances in Petri Nets 1986-Part I*, pages 168–205, London, UK, 1987. Springer-Verlag.
4. H. Boucheneb, D. Lime, and O. H. Roux. On multi-enabledness in time Petri nets. Lecture Notes in Computer Science. Springer, 2013.
5. M. Boyer and O. H. Roux. Comparison of the expressiveness of arc, place and transition time Petri nets. In *ICATPN*, volume 4546 of *Lecture Notes in Computer Science*, pages 63–82. Springer, 2007.
6. T. Chatain. *Dépliages symboliques de réseaux de Petri de haut niveau et application à la supervision des systèmes répartis.* Thèse de doctorat, Université Rennes 1, Rennes, France, November 2006.
7. T. Chatain and C. Jard. Complete finite prefixes of symbolic unfoldings of safe time Petri nets. In *ICATPN*, volume 4024 of *Lecture Notes in Computer Science*, pages 125–145. Springer, 2006.
8. J. Desel and J. Esparza. *Free Choice Petri nets.* Cambridge University Press, 1995.
9. J. Engelfriet. Branching processes of Petri nets. *Acta Informatica*, 28(6):575–591, 1991.
10. P. M. Merlin and D. J. Farber. Recoverability of communication protocols – implications of a theorical study. *IEEE Transactions on Communications*, 24, 1976.
11. P. Niebert and H. Qu. Adding invariants to event zone automata. In *FORMATS*, volume 4202 of *Lecture Notes in Computer Science*, pages 290–305. Springer, 2006.
12. P.-A. Reynier and A. Sangnier. Weak time Petri nets strike back! In *CONCUR*, volume 5710 of *Lecture Notes in Computer Science*, pages 557–571. Springer, 2009.