Timed Processes of Interval-Timed Petri Nets

Elisabeth Pelz

LACL, Université Paris-Est Créteil, France
pelz@u-pec.fr

Abstract. In this paper we use partial order semantics to express the truly concurrent behaviour of interval-timed Petri nets (ITPNs) in their most general setting, i.e. with autoconcurrency and zero duration, as studied with its standard maximal step semantics in [8]. First we introduce the notion of timed processes for ITPNs inductively. Then we investigate if the equivalence of inductive and axiomatic process semantics - true for classical Petri nets - could hold for ITPNs too. We will see that the notions of independence and immediate firing obligation seem to be antagonistic ones, and that local axioms, adequate to define processes of classical Petri nets, are not sufficient to caracterize timed Processes of IITPNs. We propose several original "global" axioms which reveal to be an effective solution. Thus we yield finally a full axiomatic definition of timed processes for ITPNs.

1 Introduction

Petri nets are an algebraic and graphical formalism, proposed by Carl Adam Petri [9], used to represent complex interactions and activities in a system, they model situations like synchronisation, sequentiality, concurrency and conflict. Classical Petri nets do not carry any time information and so they are not suitable for quantitative analysis of the performance and reliability of systems with respect to time.

Several time extensions of Petri nets have been proposed in the literature, [7,10]. In this paper, we consider Interval-Timed Petri Nets (ITPNs) with autoconcurrency which are a generalisation of Timed Petri Nets. The main feature of Interval-Timed Petri Nets is that transitions which are enabled need to start immediately their firing, and the firing lasts some time within an (integer) interval. Thus in the observation the startfiring and the endfiring of a transition are considered as two distinct events. Furthermore, in the class of ITPNs we consider here, we allow transitions to take no time (i.e. to have zero duration). This obligation of immediate firing led to the standard execution of Timed Petri Nets in (simultanous) maximal steps.

Let us state precisely the concepts which lead to the describtion of the behaviour of ITPNs under maximal step semantics, cf. [8]. Each transition has its own clock. The progress of time is managed by a global clock by means of (discrete) ticks which increment all local clocks. Inbetween two ticks, we must fire as many transitions as possible, i.e., for a maximal number of enabled transitions there are startfiring events; but also, we have to endfire every fired transition which reach its maximal duration (i.e., which must endfire) plus an arbitrary multiset of fired transitions which may endfire. The maximal multiset of events which occur between two time ticks is called a global step. In [8] such standard semantics of ITPNs in terms of firing step sequences are exhaustively defined.

By convention, we only consider ITPNs with zero duration which are well-formed, as in [8], i.e. where no infinite global step is possible.

Partial order semantics allow to describe the behaviour of concurrent systems by expressing explicitly concurrency, seen as independency. Processes are the usual partial order semantics for classical Petri nets [11,2]. They are also defined for time Petri nets [1,13,12] as well as for high level Timed Petri nets (with one-safe markings) [5]. Processes of classical Petri Nets can be defined by axiomatic definitions or by inductive ones using firing sequences, as discussed in [4].

Inspired by this approach, we define in this paper timed processes for ITPNs, inductively using firing step sequences. To our knowledge processes have never been introduced for this class. Our definitions will be coherent with the approach in [5], albeit arbitrary markings and auto-concurrency introduce new challenges. We propose a way to respect also the above quoted concepts of immediate firing obligation and global tick in the inductive definition.

But when trying to define timed processes axiomatically, some antagonism appears. Let us remind that the axiomatic definition (for classical Petri Nets) states local properties which are true for all events in the process, independently of all other events and independently of any particular cut (or marking). Now, for ITPNs, events have to satisfy global contraints too, they are no longer independent but inter-dependent and cuts (before ticks) will play the role of synchronisation barriers. In particular, each tick event depends on the set of events which precede it. We will see the limits of an axiomatization where only local properties are defined and illustrate them with an example. Then gradually the global constraints are discussed and formulated in "global" axioms. Such global axioms are something original in Petri Net semantics. We succeed to give them in first order logic without quantification on sets (or cuts). Finally we are able to present a group of local and global axioms which form a total axiomatization of timed processes.

The remaining of the paper is organized as follows: Section 2 contains formal definitions about Interval-Timed Petri Nets. In Section 3 basic definitions around partial order structures can be found. Section 4 gives an inductive definition of timed processes and Section 5 details the discussion and formulation of an axiomatic one. Section 6 presents some concluding remarks.

2 Some Definitions

Let us start to define classical Petri nets.

Definition 1 (***Petri Nets***)
A Petri net is a 3-tuple $N = (P, T, v)$ *such that*
- P *and* T *are finite sets of places and transitions respectively with* $P \cap T = \emptyset$
- $v : (P \times T) \cup (T \times P) \longrightarrow \mathbb{N}$ *is its valuation function.* □

The states of Petri nets are described by *markings* $M : P \longrightarrow \mathbb{N}$ which are represented by vectors of dimension $|P|$.

Let x be a node such that, $x \in P \cup T$. The preset of x is denoted by ${}^{\bullet}x$, with ${}^{\bullet}x = \{y \in P \cup T \mid v(y,x) > 0\}$. Similarly, $x^{\bullet}$ denotes the postset of x, with $x^{\bullet} = \{y \in P \cup T \mid v(x,y) > 0\}$. In the same manner, the preset of a net N is defined by ${}^{\bullet}N = \{x \in P \cup T \mid {}^{\bullet}x = \emptyset\}$, the postset of a net N is defined by $N^{\bullet} = \{x \in P \cup T \mid x^{\bullet} = \emptyset\}$.

Formally, a multiset U of events E is a mapping $U : E \rightarrow \mathbb{N}$, such that, for $e \in E$ the natural number $U(e)$ is called the multiplicity of e. The multiset U can be written in the extended set notation $U = \{e^{U(e)} \mid e \in E \text{ and } U(e) \neq 0\}$.

Several time extensions of classical Petri nets have been proposed to integrate temporal modeling properties, time Petri nets [7], timed Petri nets [10] and causal time Petri nets [3].

In this paper, Interval-Timed Petri Nets (ITPNs) are considered in their most general setting, i.e. allowing zero duration and autoconcurrency. ITPNs are an extension of Timed Petri Nets in which the firing duration of each transition is given within an interval. We recall shortly the definitions of [8], which contains much more details and examples.

Definition 2 $\big($***Interval-Timed Petri Nets***$\big)$
An Interval-Timed Petri Net is a 5-tuple $\mathcal{N} = (P, T, v, M_0, I)$ *such that*
- (P, T, v) *is a Petri net, called skeleton of the net* $\mathcal{N}$
- $M_0 : P \longrightarrow \mathbb{N}$ *is its initial marking*
- $I : T \longrightarrow \big[\mathbb{N}, \mathbb{N}\big]$ *is its interval function.* □

We suppose that the set of transitions is enumerated: $T = \{t_1, t_2, .., t_{|T|}\}$. The time interval associated with a transition t is given by
$I(t) = \big[sfd(t), lfd(t)\big]$, where $sfd(t)$ is called the *shortest firing duration* and $lfd(t) \geq sfd(t)$ the *longest firing duration.*
Only ITPNs are considered whose transitions have a non empty preset and postset i.e., for each transition $t \in T$ holds that $|{}^{\bullet}t| > 0$ and $|t^{\bullet}| > 0$.

In Interval-Timed Petri Nets a marking is not sufficient to describe completely the state of a net. The state must also include temporal informations. This is given by a matrix which codes the transitions clocks.

Definition 3 $\big($***State***$\big)$
A state of an ITPN $\mathcal{N} = (P, T, v, M_o, I)$ *is a pair* $S = (M, h)$ *such that*
- M *is a marking.*
- h *is a clock matrix which has* $|T|$ *rows and* d *columns s.t.* $d = \max\limits_{t_i \in T} \big(lfd(t_i) + 1\big)$.
The value $h_{i,j+1}$ *represents the number of* ***active*** *transitions* t_i *with age* j *(i.e.* ***fired*** *since* j *time ticks).* □

The set of all possible states of $\mathcal{N}$ is denoted by $States(\mathcal{N})$. The initial state of $\mathcal{N}$ is denoted $S_0 = (M_0, h_0)$ where M_0 is the *initial marking* of the skeleton and h_0 is a zero matrix, i.e. no transition is active.

2.1 Firing rules for ITPNs

D*efinition 4** $\big($Autoconcurrently enabled transitions*** $\big)$
Let N be an ITPN and $S = (M, h)$ its current state. Then a transition t is enabled at the marking M n times autoconcurrently if $\forall p \in P, \quad n \cdot v(p,t) \leqslant M(p)$. If $n = 1$ this is the usual definition of (single) enabling.
For each transition the value $E_i(M)$ tells how many times (at most) transition t_i can be fired autoconcurrently at marking M. Thus $E_i(M) = n_i$ if
$\forall p \in P, \quad (n_i \cdot v(p,t_i) \leqslant M(p)$ *and* $\exists p \in P, \quad (n_i + 1) \cdot v(p,t_i) > M(p))$. $\square$

When transitions may fire, firing starts *immediately*; this is done by removing input tokens from the preplaces of the chosen transitions. A startfired transition t stays *active* for some time delay in between its associated time interval $\big[sfd(t), lfd(t)\big]$, until it may or must *endfire* by delivering the output tokens to its postplaces.
Three types of events are distinguished : *startfire, endfire and tick events.* The effect of each of these events on the state of an ITPN is given below.

D*efinition 5** $\big($State change rules*** $\big)$
Let $\mathcal{N}$ be an ITPN and (M, h) its current state.

1. **Startfire events :** *A startfire event, denoted by $[\mathtt{t_i}$, may occur immediately, even up to n times, if t_i is enabled at M, resp. if $E_i(M) = n$. For each occurrence of $[\mathtt{t_i}$ the needed input tokens of t_i are removed from their preplaces, the clock associated with t_i will count this occurrence by incrementing the number $h_{i,1}$.*
   $(M, h) \xrightarrow{[\mathtt{t_i}} (M', h')$ *with* $M' = M - \sum_{p \in {}^{\bullet}t_i} v(p, t_i)$ *and* $h'_{i,1} = h_{i,1} + 1$.
   There may be conflicts between enabled transitions, and the way they are solved is arbitrary. Tick events may not occur when there are still enabled transitions, i.e., if for some i, $E_i(M) > 0$.
2. **Endfire events :** *An endfire event, denoted by $\mathtt{t_i}\rangle$ must occur (even n times) if the clock associated with some t_i reach the upper bound of its associated interval i.e. $h_{i,j+1} \geq 1$ with $j = lfd(t_i)$. An endfire event $\mathtt{t_i}\rangle$ may occur if there is an active transition t_i with age in $[sfd(t_i), lfd(t_i)[$. The corresponding $h_{i,j+1}$ is then decremented.*
   $(M, h) \xrightarrow{\mathtt{t_i}\rangle} (M', h')$ *if* $\sum_{sfd(t_i) \leqslant j \leqslant lfd(t_i)} h_{i,j+1} > 1$ *with*
   $M' = M + \sum_{p \in t_i^{\bullet}} v(t_i, p)$ *and* $h'_{i,j+1} = h_{i,j+1} - 1$ *for some j with* $h_{i,j+1} > 0$.

   If this new state enables some transitions, one or more startfire events must then occur, and if endfire events(of zero duration) must occur in the sequel, they have to be handled, and so on.
3. **Tick events :** *A tick event, denoted by $\checkmark$, is enabled once neither a startfire event nor a must endfire event have to occur. The tick event increments the clocks for all active transitions and models the passing of time.*
   $(M, h) \xrightarrow{\checkmark} (M', h')$ *with* $M' = M$ *and for all i holds if*
   $$E_i(M) = 0 \quad \text{and} \quad h_{i,lfd(t_i)+1} = 0 \quad \text{then} \quad h'_{i,j} = \begin{cases} h_{i,j-1} & \text{if } 1 < j \leqslant d \\ 0 & \text{if } j = 1 \end{cases}. \qquad \square$$

The ITPN $\mathcal{N}$ presented in Fig.1. is used as a running example.

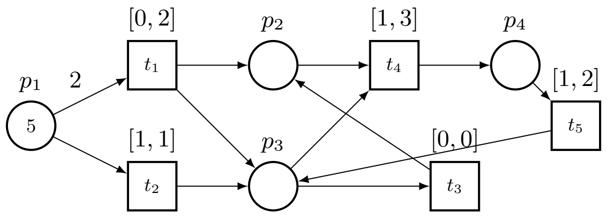

Fig. 1: ITPN $\mathcal{N}$

The condition of the occurrence of a tick event ensures that what have occurred since the previous tick event (i.e. in between two ticks) is maximal. In the case of transitions which may late zero time, which is allowed in the net class considered here, the notion of start firing event which need to occur "immediately", means "before the next tick". There is no time scale within zero time. The following definition will precise the notion of *global step* which happens in between two ticks and where multisets of startfire and endfire events will alternate until nothing more need to occur.

We only consider *wellformed* ITPNs in this article. They ensure to have always only a finite number of events which appear between two ticks. If there is no firing sequence of transitions of possible zero duration which increases a marking, the ITPN is wellformed. This property is decidable on the subnet restricted to transitions whose sfd is zero.

All results given here could be easily extended to ITPNs which are not necessarily wellformed: they allow infinite global steps where no tick event can follow. Thus such an infinite global step would be the "end" of a step firing sequence. We just like to limite the considerations here to the standard wellformed case.

2.2 Maximal Step Semantics for ITPNs

The executions of wellformed ITPNs with zero duration and autoconcurrency under maximal step semantics are given by the so called *firing step sequence* as defined in [8], where a *firing step sequence* is an alternating sequence of *globalsteps* and ticks.

A *globalstep* is a multiset of firing events of an ITPN in between two tick events, it consists on two principal multisets, the first one is called *Endstep* and the second one is called *Iteratedstep*. So a firing step is a triplet $(Endstep, Iteratedstep, tick)$, or a couple $(globalstep, tick)$.

An *Endstep* at state S contains all endfire events which must occur at S and an arbitrary multiset of endfire events which may occur at S. At the beginning, at initial state S_0, it is always empty as no transition has startfired. In the following steps, it can be empty; then the whole global step can be empty and one tick event follows immediately the previous one. An *Iteratedstep* is an alternating sequence of multisets of startfire events (not necessarily maximal) and multisets of endfire events with zero duration (containing all must endfire ones). The alternation ends if neither a transition is enabled nor an endfire event with zero duration must occur, thus the iterated step is maximal and finite. This situation happens because of the wellformedness.

Thus a firing step sequence σ of length n is given by
$\sigma = S_0 \xrightarrow{globalstep_1} \tilde{S}_0 \xrightarrow{\checkmark} S_1 \xrightarrow{globalstep_2} \tilde{S}_1 \xrightarrow{\checkmark} S_2 \cdots S_{n-1} \xrightarrow{globalstep_n} \tilde{S}_{n-1} \xrightarrow{\checkmark} S_n$.
If in $\tilde{S}_{n-1} = (\tilde{M}_{n-1}, \tilde{h}_{n-1})$ no transition is active, i.e. if $\tilde{h}_{n-1}$ is the Zero-matrix, then we have a deadlock , and the last tick and S_n do not exist.

The following example illustrates firing steps.

***E*xample 1** *Consider the ITPN $\mathcal{N}$, one possible initial firing step from its initial state S_0 is given in the sequel.*

The first Endstep is necessarily empty (Endstep$_1 = \emptyset$). Then, suppose that we fire two times t_1 and one time t_2, i.e. $\{[\mathtt{t_1}^2, [\mathtt{t_2}\}$ then we choose to endfire t_1 at zero duration, i.e. $\{\mathtt{t_1}\rangle\}$, after that we fire $\{[\mathtt{t_4}\}$, no further startfire event is possible now, so a tick event is executed. The first iterated step is the union of all these multisets :
Iteratedstep$_1 = \{[\mathtt{t_1}^2, [\mathtt{t_2}\} \uplus \{\mathtt{t_1}\rangle\} \uplus \{[\mathtt{t_4}\} = \{[\mathtt{t_1}^2, [\mathtt{t_2}, \mathtt{t_1}\rangle, [\mathtt{t_4}\}$.

Thus a possible first firing step of $\mathcal{N}$ is $\big(\{\}, \{[\mathtt{t}_1^2, [\mathtt{t_2}, \mathtt{t_1}\rangle, [\mathtt{t_4}\}, \checkmark\big)$. □

3 True concurrent semantics

In this paper we plan to study the behaviour of ITPNs without sequentializing the observation. Thus we will use partial order semantics to express true concurrency and in particular, nonsequential processes [6,2,11].

Processes have been defined and investigated for classical Petri nets and for some other net classes like **time** Petri Nets [13,1,12]. The only known contribution on process semantics of a **timed** Petri Net class is [5], but in a context of high level nets with one-safe markings. Arbitrary markings, zero durations of events and auto-concurrency give us new challenges. Also, no axiomatic approach exists until now for time or timed Petri nets.

3.1 Partial order structures

Concurrent runs or executions of an ITPN are usually represented by condition/event nets where all arcs have an arc weight 1.

$N' = (B, E, G)$ is a *condition/event net* if $B \cap E = \emptyset$ and $G \subseteq (B \times E) \cup (E \times B)$. The places of B are called *conditions* and the transitions of E are called *events.*

A *causal net* is a condition/event net $N' = (B, E, G)$ such that

- for every $b \in B$, $|\,^\bullet b| \leqslant 1$ and $|b^\bullet| \leqslant 1$,
- G^*, the transitive closure of G is acyclic,

– N' is finitly preceded, i.e., for every $x \in B \cup E$ the set $\{y \mid (y, x) \in G^*\}$ is finite.

Causal nets do not allow any branching at conditions. In a causal net $N' = (B, E, G)$, the transitive closure of the flow relation G is acyclic and therefore a partial order. We call it the *precedence relation* and denote it by $\prec$. The symbol $\preceq$ denotes the reflexive and transitive closure of G.

A *homomorphism* ϕ is a mapping that preserves the nature of nodes and the environment of events. A homomorphism is used to connect conditions and events of a *causal net* to places and transitions of the executed net whose behaviour is observed.

A *chain* C of a causal net is a set of totally ordered events, i.e., $\mathrm{C} \subset E$ and $\forall e \in \mathrm{C}\ \forall e' \in \mathrm{C}\ \big((e' \preceq e) \vee (e \preceq e')\big)$. It can be seen as a sequence of events that occurred during the run of the system.

A set AC of nodes of a causal net is an *antichain* if $\forall x \in \mathrm{AC}\ \forall x' \in \mathrm{AC}\ \ (\neg(x \prec x') \wedge \neg(x' \prec x))$. An antichain AC is a *maximal antichain* or a *cut* if $\forall x \notin \mathrm{AC}$ the union $\mathrm{AC} \cup \{x\}$ is not an antichain.

Note that usually, cuts are considered restricted to conditions or restricted to events. In particular, a cut restricted to conditions is called *cut of conditions* or *B-cut.* Note that each *B-cut* of a process of a classical Petri Net represent a possible marking that may occur during the concurrent execution for some observer.

4 Process semantics for ITPNs

A **timed process** of an ITPN $\mathcal{N}$ will be defined as a pair (N', ϕ) where N' is a *causal net* and ϕ a homomorphism which labels the causal net with information from the ITPN $\mathcal{N}$. The set of *clock labels* is introduced to capture information about time elapsed since a transition is active. It is defined by

$$CL = \{(t, j) \mid t \in T \text{ and } j \leqslant \mathit{lfd}(t_i)\} \text{ and } P \cap CL = \emptyset.$$

A clock label (t, j) means that t is active and has age j.

The following set of *firing events* denoted by $\mathcal{FE}$ will label the events:

$$\mathcal{FE} = \{[\mathtt{t} \mid t \in T\} \cup \{\mathtt{t}\rangle \mid t \in T\} \cup \{\checkmark\} \text{ and } P \cap \mathcal{FE} = \emptyset.$$

Thus in a causal net where conditions are labeled in $P \cup CL$ a *B-cut* is able to represent a time-state (M,h). Note that for any set $B' \subseteq B$ the image $\phi(B')$ defines a *multi-set* of labels.

4.1 Inductive definition

Let $\mathcal{N} = (P, T, v, M_0, I)$ be an ITPN. A **timed process** π **of** $\mathcal{N}$ is constructed along a possible firing step sequence σ of $\mathcal{N}$, whose length is n_o, as follows.

We construct successively labeled causal nets $\pi_i = (N'_i, \phi_i) = (B_i, E_i, G_i, \phi_i)$ where $\phi_i : B_i \cup E_i \to (P \cup CL) \cup \mathcal{FE}$ by induction on i by using three **Add**-procedures given below for the creation of events. The $i-$th induction step corresponds to the $i-$th firing step of a σ. We stop if $\pi = (N', \phi) = \pi_{n_o}$.

The sets B_{CL} and B_P will be the sets of conditions whose postset is currently empty and which are labeled by clock labels and by places respectively.

Base of induction $\mathbf{i = 0}$: ${}^\bullet\pi_0 = B_0$ will be a set of conditions with $\phi_0 : B_0 \to P$ representing the initial marking such that
$\forall p \in P \ \ |{\phi_0}^{-1}(p) \cap B_0| = M_0(p)$. We set $E_0 = G_0 = \emptyset$, $B_P = B_0$ and $B_{CL} = \emptyset$.

Hypothesis: Let $\mathbf{n \geqslant 1}$. We suppose that $\forall i < n$, $\pi_i = (B_i, E_i, G_i, \phi_i)$ has been constructed and the current B_{CL} and B_P are known.

Induction step $\mathbf{i = n}$: We start by setting $B_i = B_{i-1}, E_i = E_{i-1}, G_i = G_{i-1}$ and $\phi_i = \phi_{i-1}$. Then π_i is constructed as follows:

a) (*Treatment of the first Endstep of the current globalstep*)
For each condition $b \in B_{CL}$:
- If $\phi_i(b) = (t, j)$ for some t with $j = \mathit{lfd}(t)$ then $\mathbf{Add}\big(b, \mathtt{t}\rangle, i\big)$.
- If $\phi_i(b) = (t, j)$ for some t with $\mathit{sfd}(t) \leqslant j < \mathit{lfd}(t)$ then $\mathbf{Add}\big(b, \mathtt{t}\rangle, i\big)$ or do nothing.

b) (*Treatment of the Iteratedstep of the current globalstep*)
b.1) (*Treatment of Startfirings*)
If there exists a set $B' \subseteq B_P$ with $\phi_i(B') = {}^\bullet t$ for some $t \in T$, then $\mathbf{Add}\big(B', [\mathtt{t}, i\big)$.
Repeat b.1) or goto to b.2) .
b.2) (*Treatment of an Endstep*)
For each condition $b \in B_{CL}$:
- If $\phi_i(b) = (t, 0)$ for some t with $\mathit{lfd}(t) = 0$, then $\mathbf{Add}\big(b, \mathtt{t}\rangle, i\big)$.
- If $\phi_i(b) = (t, 0)$ for some t with $\mathit{sfd}(t) = 0$ and $\mathit{lfd}(t) \neq 0$, then $\mathbf{Add}\big(b, \mathtt{t}\rangle, i\big)$ or do nothing.

(*Maximality of the globalstep*)
Repeat step b) until $\forall t \in T : \ {}^\bullet t \nsubseteq \phi(B_P))$ (i.e. until no startfire event is possible) , then go to c).

c) (*Treatment of a Tickevent*)
If $B_{CL} \neq \emptyset$ then $\mathbf{Add}\big(\checkmark, i\big)$. □

End (*If $i = n_o$*)

The three **Add**-procedures are as follows:
The startfire event creation $\mathbf{Add}\big(B', [\mathtt{t}, i\big)$:
- We add an event e with $\phi_i(e) = [\mathtt{t}$: $E_i = E_i \cup \{e\}$.
- We add arcs $G_i = G_i \cup \{(b, e) | b \in B'\}$.
- We add a condition b' with $\phi_i(b') = (t, 0)$: $B_i = B_i \cup \{b'\}; B_{CL} = B_{CL} \cup \{b'\}$.
- We add an arc $G_i = G_i \cup (e, b')$ and reset $B_P = B_P \setminus B'$.

The endfire event creation $\mathbf{Add}\big(b, \mathtt{t}\rangle, i\big)$:
- We add an event e with $\phi_i(e) = \mathtt{t}\rangle$: $E_i = E_i \cup \{e\}$.
- We add an arc $G_i = G_i \cup (b, e)$ and redefine $B_{CL} = B_{CL} \setminus \{b\}$.

– For each $p \in t^\bullet$, we add $v(t,p)$ conditions $B' = \{b'_1, .., b'_{v(t,p)}\}$ with $\phi_i(b') = p$ for all $b' \in B'$: $B_i = B_i \cup B'$ and $B_P = B_P \cup B'$.
– We add arcs $G_i = G_i \cup \{((e,b')|b' \in B'\}$.

The tick event creation $\mathbf{Add}(\checkmark, i)$:
– We add an event e with $\phi_i(e) = \checkmark$: $E_i = E_i \cup \{e\}$.
– We add arcs $G_i = G_i \cup \{(b,e)|b \in B_{CL}\}$.
– For each $b \in B_{CL}$, if $\phi_i(b) = (t,j)$ for some t and j, then we add a condition b' with $\phi_i(b') = (t, j+1)$: $B_i = B_i \cup \{b'\}$ and an arc $G_i = G_i \cup (e,b')$.
– We redefine $B_{CL} = e^\bullet$.

We observe that c) happens at each step because of the wellformedness of the executed ITPN. If $B_{CL} = \emptyset$ then a deadlock appeared; otherwise a tick is added. A process construction stops after the creation of some tick event, except for deadlock. Thus global steps are fully included. It is easy to see, that for each i the cut $\pi_i^\bullet$ is a *B-cut* and represents the time-state S_{i+1} reached after the execution of the considered firing step sequence of length i, respectively $\tilde{S}_{n_0-1}$ in the case of a deadlock after n_0 global steps.

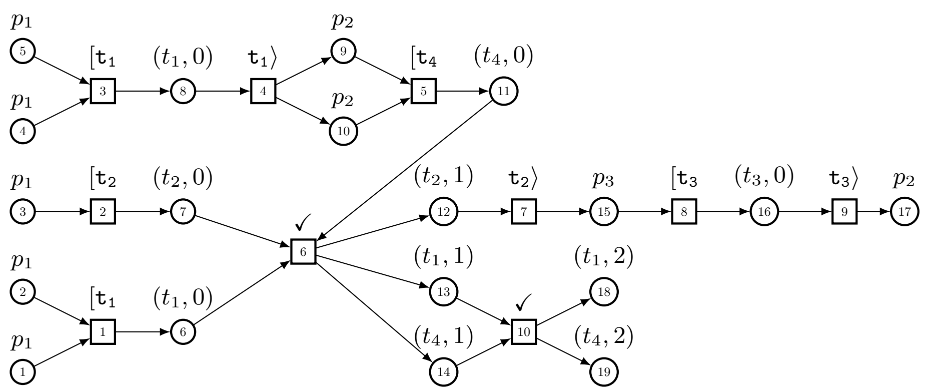

Fig. 2: An arbitrary process of the ITPN $\mathcal{N}$ of Fig.1.
We use abreviation as follows: $\textcircled{i}$ Denotes condition b_i and $\boxed{j}$ denotes event e_j.

4.2 Axiomatic definition and how to overcome its limits

We start by proposing an axiomatization by local properties of events as processes of a classical Petri Nets have been axiomatized, for instance in [4].
The causal net $\pi = (N', \phi)$ is an **timed evolution of** $\mathcal{N}$ if

$\phi : B \cup E \to (P \cup CL) \cup \mathcal{FE}$ is a homomorphism verifying
– $\forall b \in B,\ \phi(b) \in P \cup CL$ and $\forall e \in E,\ \phi(e) \in \mathcal{FE}$ (coherence of labeling).
– ${}^\bullet\pi \subseteq B$ and $\forall p \in P,\ |\phi^{-1}(p) \cap {}^\bullet\pi| = M_0(p)$ (the initial marking).
– For each event e of the causal net N' it holds :
  - **Case 1 :** If $\phi(e) = [\mathtt{t}$ for some $t \in T$ then
    * $\forall p \in P \quad |\phi^{-1}(p) \cap {}^\bullet e| = v(p,t)$ and $|e^\bullet| = 1$ with $\phi(e^\bullet) = \{(t,0)\}$

- **Case 2:** If $\phi(e) = \mathtt{t}\rangle$ for some $t \in T$ then
  - $*$ $|{}^{\bullet}e| = 1$ and $\phi({}^{\bullet}e) = \{(t,j)\}$ for some $j \in [sfd(t), lfd(t)]$ and
  - $*$ $\forall b \in e^{\bullet}\ \phi(b) \in P$ and $\forall p \in P\ \ |\phi^{-1}(p) \cap e^{\bullet}| = v(t,p)$.
- **Case 3:** If $\phi(e) = \checkmark$ then
  - $*$ $\forall b \in {}^{\bullet}e \cup e^{\bullet}\quad \phi(b) \in CL$ and
  - $*$ $\forall b \in {}^{\bullet}e\quad \phi(b) = (t,j))$ for some t and some $j < lfd(t)$ and
  - $*$ $\forall t \in T\quad \forall j \in [0,d]\ |\phi^{-1}((t,j)) \cap {}^{\bullet}e| = |\phi^{-1}((t,j+1)) \cap e^{\bullet}|$

These axioms define especially local properties of events in the same way as axioms of processes for classical Petri Nets, i.e., they ensure that each event has a correct pre- and postset of conditions with respect to the firing rule. Only the initial cut and the final one are evoked. It is evident that each event of an inductively defined process satisfies clearly the corresponding axiom.

First let us state the following sentence.

Proposition 1 *There are evolutions which are not processes.*

Proof. An evolution of the net N of Fig.1 is given in Fig.3. It respects all points of the axiomatic definition but does not correspond to any firing step sequence of N. We can see in this example that tick event e_2 is not global as it should be, and may only occur when neither a startfire event nor a must endfire event is possible. In particular, e_3 and e_4 are independent from e_2, thus conditions b_8 and b_9 are also independent from e_2 instead of entering it. These axioms also allow infinite evolutions, and we could have added an axiom like $[\pi^{\bullet} \neq \emptyset$ and $\forall b \in B\ \ \exists x \in \pi^{\bullet}\ \ b \leq x]$ to ensure that π is finite. But as finiteness will be a consequence of the axioms adjoined in the sequel, we omit it here.

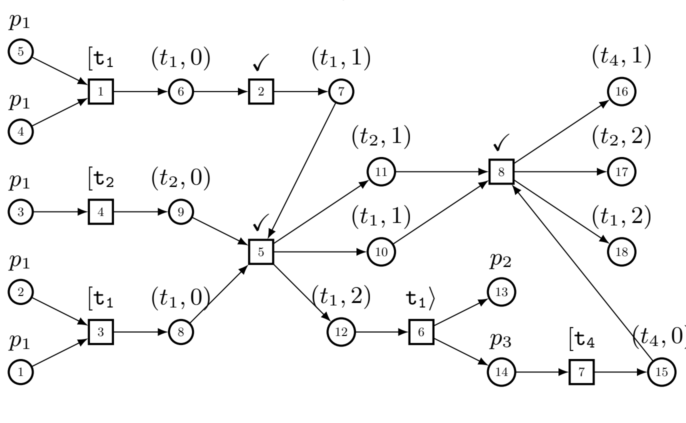

Fig. 3: An evolution which is not a process of running example.

Therefore, with the given axiomatic definition we are unable to avoid some partial orders that violate important properties like the fact that tick events have to be global and have to form a chain. Let us try to formulate supplementary axioms about non local properties for tick events:

Globality axioms:
- (a) $\forall e\ \forall e'\ \ \big((\phi(e) = \checkmark \wedge \phi(e') = \checkmark) \Rightarrow\ \ (e = e' \vee e' \prec e \vee e \prec e')\big)$
- (b) $\forall e\ \forall e'\ \ \big((\phi(e) = \checkmark \wedge \phi(e') = \checkmark \wedge e' \prec e)) \Rightarrow (\forall b \in {}^{\bullet}e\ \ e' \prec b)\big)$

The axiom (a) ensures that all tick events form a chain in the partial order, and (b) that all conditions entering later tick events are necessarily greater than other preceding tick events, thus in particular greater than its potential direct predecessor tick. In particular, for the running example, point (b) makes impossible to creat a "partial" tick event like e_2 in Fig.3, as b_9 and b_8 entering e_5 are not comparable to (and not greater than) e_2.

Finally global axioms about maximality and concerning the final cut have to be defined.

Final cut axioms:

– (d) $\forall t \in T \;\; \exists p \in {}^\bullet t \;\; |\phi^{-1}(p) \cap \pi^\bullet| < v(p,t)$
– (e) $\forall t \in T \;\; \forall b \;\; (\phi(b) = (t, lfd(t)) \Rightarrow |b^\bullet| = 1)$
– (f) $\forall x \in \pi^\bullet \; \phi(x) \in P \;\; \vee \;\; \exists e \; \big(\, (\phi(e) = \checkmark \wedge \forall x \; (e \prec x \Rightarrow \phi(x) \in CL) \wedge$
$\forall b \notin e^\bullet \; (\phi(b) \in CL) \Rightarrow b^\bullet \neq \emptyset) \, \big)$

As by (d) no startfire event is possible at the final cut $\pi^\bullet$, availible tokens (P labeled conditions) are maximally used for startfire events and thus the obligation of startfiring, up to some choice, is satisfied.

By axiom (e) must endfire events must occur.

Axiom (f) ensures that either (case 1) all elements in the final cut are place labeled, which together with (d) means that the process ends by an deadlock; or (case 2) there is a last tick event whose postset are clock labeled conditons in the final cut $\pi^\bullet$ and all other clock labeled conditons have a successor; i.e., they enter in an endfiring event, or they enter in a tick event which is by axiom (b) the appropriate one and not a later one.

We may conclude that what happens between two tick events is a global step and axioms (d) and (e) together ensure its maximality. Axiom (f) also implies the finiteness of the process. Finally the finite cut correspond to the time state reached after the execution of all events of the evolution.

The initial cut ${}^\bullet\pi$ and the final cut $\pi^\bullet$ are the only sets of nodes evoked in the given axioms; they are just used like constant sets. Thus we have successfully avoided to use second order quantification over sets - representing intermediate *B-cut*s - in all proposed axioms.

As consequence of these observations we obtain the desired result.

Proposition 2 *Let $\mathcal{N}$ be a wellformed ITPN. Then the class of timed evolutions of $\mathcal{N}$ which also satisfy the axioms (a) to (f) is the same as that of timed processes of $\mathcal{N}$ defined inductively.*

5 Conclusion

In this paper we investigate ITPNs in their most general setting, i.e., with autoconcurrency and with zero duration. In a previous paper their usual maximal step semantics were introduced [8], in terms of firing step sequences.

The goal of the present article is to present their truly concurrent behaviour. Thus first, timed processes of ITPNs have been defined inductively along firing step sequences. Then the possibility of defining these processes in an axiomatic way too are studied. Our first attempt was to propose local axioms, similar to the way processes of classical Petri Nets are defined axiomatically, obtaining the so called timed evolutions. Then we stated and illustrated the fact that some timed evolutions do not correspond to any firing step sequence and therefore they cannot be timed processes.

Several supplementary "global" axioms are gradually formulated and discussed. They are a novelty when defining processes, but the price to pay to capture global timing and firing constraints.

We succeed to give a full axiomatization of timed processes totally compatible with the firing step semantics.

6 Acknowlegment

My thanks go to Raymond Devillers for his attentive reading of divers versions and a lot of pertinent remarks and suggestions.

References

1. T. Aura and J. Lilius. Time processes for time petri-nets. In *Proc. of 18th International Conference ICATPN '97, LNCS*, volume 1248, pages 136–155. Springer, 1997.
2. E. Best and C. Fernández. *Nonsequential Processes - A Petri Net View*, volume 13 of *EATCS Monographs on Theoretical Computer Science*. Springer, 1988.
3. C. Bui Thanh, H. Klaudel, and F. Pommereau. Petri nets with causal time for system verification. *Electr. Notes Theor. Comput. Sci.*, 68(5):85–100, 2002.
4. E.Best and R.Devillers. Sequential and concurrent behaviour in petri net theory. *Theoretical Computer Science*, 55(1):87–136, 1987.
5. H. Fleischhack and E. Pelz. Hierarchical Timed High Level Nets and their Branching Processes. In *Proc. of 26th International Conference ICATPN'03, LNCS* , volume 2679, pages 397–416. Springer, 2003.
6. U. Goltz and W. Reisig. The non-sequential behavior of petri nets. *Information and Control*, 57(2/3):125–147, 1983.
7. P. Merlin. *A Study of the Recoverability of Communication Protocols.* PhD thesis, Irvine, 1974.
8. E. Pelz, A. Kabouche, and L. Popova-Zeugmanm. Interval-timed petri nets with auto-concurrent semantics and their state equation. In *Proc. of International Workshop on Petri Nets and Software Engineering (PNSE'15), CEUR Workshop, http://ceur-ws.org/Vol-1372*, pages 245–265, 2015.
9. C. A. Petri. Fundamentals of a Theory of Asynchronous Information Flow. In *IFIP Congress*, 1962.
10. C. Ramchandani. Analysis of Asynchronous Concurrent Systems by Timed Petri Nets. *Project MAC-TR 120, MIT*, February 1974.
11. P. H. Starke. Processes in petri nets. *Elektronische Informationsverarbeitung und Kybernetik*, 17(8/9):389–416, 1981.
12. V. Valero Ruiz, D. de Frutos-Escrig, and F. Cuartero. Timed processes of timed petri nets. In *Proc. of 16th International Conference, ICATPN '95, LNCS*, volume 616, pages 490–509. Springer, 1995.
13. J. Winkowski. Algebras of processes of timed petri nets. In *Proc. of 5th International Conference CONCUR '94, LNCS*, volume 836, pages 194–209. Springer, 1994.