Combining Free Choice and Time in Petri Nets

Sundararaman Akshay, Loïc Hélouët, Ramchandra Phawade

Combining Free Choice and Time in Petri Nets

S. Akshay
IIT Bombay
Email: akshayss@cse.iitb.ac.in

Loïc Hélouët
INRIA Rennes
Email: loic.helouet@inria.fr

Ramchandra Phawade
IIT Bombay
Email: ramchandra@cse.iitb.ac.in

***Abstract*—Time Petri nets (TPNs) (Merlin 1974) are a classical extension of Petri nets with timing constraints attached to transitions, for which most verification problems are undecidable. We consider TPNs under a strong semantics with multiple enabling of transitions. We focus on a structural subclass of unbounded TPNs, where the underlying untimed net is free choice, and show that it enjoys nice properties under a multi-server semantics. In particular, we show that the questions of fireability (whether a chosen transition can fire), and termination (whether the net has a non-terminating run) are decidable for this class. We then consider the problem of robustness under guard enlargement (Puri et al. 2000), i.e., whether a given property is preserved even if the system is implemented on an architecture with imprecise time measurement. This question was studied for TPNs in (Akshay et al. 2016), and decidability of several problems was obtained for bounded classes of nets. We show that robustness of fireability is decidable for unbounded free choice TPNs with a multi-server semantics.**

I. Introduction

Modern systems are composed of several distributed components that work in real-time to satisfy a given specification. This makes them difficult to reason about manually and encourages the use of formal methods to analyze them automatically. This in turn requires the development of models that capture all the features of a system and still allow efficient algorithms for analysis. Further, to bring formal models closer to real world implementions, it is important to design robust models, i.e., models that preserve their behavior or at least some important properties under imprecise time measurement.

In this paper, we consider Petri nets extended with time constraints. These models have been used for modeling real-time distributed systems, but for timed variants of Petri nets, many basic problems are usually undecidable or algorithmically intractable. Our goal is to consider structural restrictions which allow to model features such as unbounded resources as well as time deadlines while remaining within the realm of decidability and satisfying some robustness properties.

Time Petri nets (TPNs) [16] are a classical extension of Petri nets in which time intervals are attached to transitions and constrain the time that can elapse between the enabling of a transition and its firing date. In such models, the basic verification problems considered include: **reachability**, i.e., whether a particular marking (or a configuration) can be reached in the net, **termination**, i.e., whether there exists an infinite run in the net, **boundedness**, whether there is a bound on the number of tokens in the reachable markings, and **fireability**, i.e., whether a given transition is fireable in some execution of the net. It turns out that all these basic problems are in general undecidable [15] for TPNs, though they are decidable for the untimed version of Petri nets. The main reason is that TPNs are usually equipped with an urgent semantics: when the time elapsed since enabling of a transition reaches the maximal value of its interval, a transition of the net *has to fire*. This semantics breaks monotony (the behaviors allowed from a marking, and from a larger marking can be completely different). Indeed, with a TPN, one can easily encode a two-counter machine, yielding undecidability of most of verification problems (see [15], [19] for such an encoding). Decidability can be obtained by restricting to the *subclass* of *bounded TPNs*, for which the number of tokens in all places of accessible markings is bounded by some constant. Another way to obtain decidability is to weaken the semantics [19], or restrict the use of urgency [2].

Another important problem in this setting is ensuring **robustness**. Robustness deals with preservation of properties in systems that are subject to imprecision of time measurement. The main motivation for considering this issue is to address the idealized representation of time in formal methods, which assumes an unrealizable precision in time measurement that cannot be guaranteed by real implementations. Robustness has been considered extensively for timed automata since [17], and more recently for TPNs [3], but decidability results are only obtained in a bounded-resource setting.

The definition of the semantics plays an important role both to define the expressive power of a model and to obtain decidability results. When considering unbounded nets, where multiple (and in fact unboundedly many) tokens may be present at every place, one has to fix a policy on whether and how multiply-enabled transitions are handled. And this becomes even more complicated when real-time constraints are considered. Several possible variants for multiple enabling semantics have been considered in [8], [10] and later in [7]. In this paper, we fix one of the variants and consider TPNs equipped with a multi-enabling urgent semantics, which allows to start measuring elapsed time from every occurrence of a transition enabling. This feature is particularly interesting: combined with urgency, it allows for instance to model maximal latency in communication channels. We adopt a semantics where time is measured at each transition's enabling, and with urgency, i.e. a discrete transition firing has to occur if a transition has been enabled for a duration that equals the upper bound in the time interval attached to it. Obviously, with this semantics, counter machines can still be encoded, and undecidability follows in general.

We focus on a structural restriction on TPNs in which the underlying net is free choice, and call such nets *free choice TPNs*. Free choice Petri nets have been extensively studied in the untimed setting [13] and have several nice properties from a decidability and a complexity-theoretic point of view. In this class of nets, all occurrences of transitions that have a common place in their preset are enabled at the same instant. Such transitions are said to belong to a cluster of transitions. Thus, with this restriction, a transition can only prevent transitions from the same cluster to fire, and hence only constrain firing times of transitions in its cluster. Further, we disallow forcing of instantaneous occurrence of infinite behaviors, that we call forced 0-delay firing sequences in our nets. This can easily be ensured by another structural restriction forbidding transitions or even loops in TPNs labeled with $[0, 0]$ constraints.

Our main results are the following: we show that for a free choice TPN $\mathcal{N}$ under the multiple-enabling urgent semantics, and in the absence of forced 0-delay infinite firing sequences, the problem of fireability of a transition and of termination are both decidable. The main idea is to show that, after some pre-processing, we can reduce these problems to corresponding problems on the underlying untimed free choice PN. More precisely, we show that every partially-ordered execution of the underlying untimed PN of a TPN $\mathcal{N}$ can be simulated by $\mathcal{N}$, i.e., it is the untimed prefix of a timed execution of $\mathcal{N}$. To formalize this argument we introduce definitions of (untimed and timed) causal processes for unbounded TPNs, which is a second contribution of the paper.

Finally, we address the problem of robustness for TPNs which has previously been considered in [3]. We show that robustness of fireability wrt guard enlargement, i.e., whether there exists a $\Delta > 0$ such that enlarging all guards of a TPN by Δ preserves the set of fireable transitions, is decidable for free choice TPNs. Up to our knowledge, this is the first decidability result on robustness for a class of unbounded TPNs.

Related work. Verification, unfolding, and extensions of Petri nets with time have been considered in many works. Another way to integrate time to Petri nets is to attach time to tokens, constraints to arcs, and allow firing of a transition iff all constraints attached to transitions are satisfied by at least one token in each place of its preset. This variant, called Timed-arc Petri nets, enjoys decidability of coverability [1], but cannot impose urgent firing, which is a key issue in real-time systems. In TPNs, [19] propose a weak semantics for TPNs, where clocks may continue to evolve even if a transition does not fire urgently. With this semantics, TPNs have the same expressive power as untimed Petri nets, again due to lack of urgency, which is not the case in our model.

Recently, [2] considers variants of time and timed-arc Petri nets with urgency (TPNUs), where decidability of reachability and coverability is obtained by restricting urgency to transitions consuming tokens only from bounded places. This way, encoding of counter machines is not straightforward, and some problems that are undecidable in general for time or timed-arc Petri nets become decidable. The free choice assumption in this paper is orthogonal to this approach and it would be interesting to see how it affects decidability for TPNUs.

Partial-order semantics have been considered in the timed setting: [20] defines a notion of timed process for timed-arc Petri nets and [21] gives a semantics to timed-arc nets with an algebra of concatenable weighted pomsets. However, processes and unfoldings for TPNs have received less attention. An initial proposal in [5] was used in [9] to define equivalences among time Petri nets. Unfolding and processes were refined by [11] to obtain symbolic unfoldings for *safe* Petri nets. The closest work to ours is [12], where processes are defined to reason about the class of free choice *safe* TPNs. However, this work does not consider unbounded nets, and focuses more on semantic variants wrt time-progress than on decidability or robustness issues.

The paper is organized as follows: Section II introduces notations and defines TPNs with multi-enabling semantics. Section III defines processes for these nets. Section IV introduces the subclass of free choice TPNs and relates properties of untimed and time processes. Section V proves decidability of fireability and termination for FC-TPNs and Section VI addresses robustness of fireability. Section VII discusses our assumptions and deals with issues related to decidability of other problems in FC-nets. Due to space constraints, some proofs are only sketched, and details can be found in [4].

II. Multi-enabledness in TPNs

Let Σ be a finite alphabet, Σ^* be the set of finite words over Σ. For a pair of words $w, w' \in \Sigma^*$, we will write $w \leq w'$ iff $w' = w.v$ for some $v \in \Sigma^*$. Let $\mathbb{N}, \mathbb{Q}_{\geq 0}, \mathbb{R}_{\geq 0}$, respectively, denote the sets of naturals, non-negative rationals and real numbers. An interval I of $\mathbb{R}_{\geq 0}$ is a $\mathbb{Q}_{\geq 0}$-interval iff its left endpoint belongs to $\mathbb{Q}_{\geq 0}$ and right endpoint belongs to $\mathbb{Q}_{\geq 0} \cup \{\infty\}$. Let $\mathcal{I}$ denote the set of $\mathbb{Q}_{\geq 0}$-intervals of $\mathbb{R}_{\geq 0}$. For a set X of (clock) variables, a valuation v for X is a mapping $v : X \to \mathbb{R}_{\geq 0}$. Let $v_0(X)$ be the valuation which assigns value 0 to each clock in X. For any $d \in \mathbb{R}_{\geq 0}$, the valuation $v + d$ is : $\forall x \in X,\ (v + d)(x) = v(x) + d$.

A Petri net (PN) $\mathcal{U}$ is a tuple $\mathcal{U} = (P, T, F)$, where P is a set of places, $T = \{t_1, t_2, \ldots, t_K\}$ is a set of transitions, and $F \subseteq (P \times T) \cup (T \times P)$ is a flow relation. A marking M is a function $P \to \mathbb{N}$ that assigns a number of tokens to each place. We let M_0 denote an initial marking for a net. For any $x \in P \cup T$ (called a *node*) let ${}^\bullet x = \{y \in P \cup T \mid (y, x) \text{ in } F\}$ and $x^\bullet = \{y \in P \cup T \mid (x, y) \text{ in } F\}$. For a transition t, we will call ${}^\bullet t$ the preset of t, and $t^\bullet$ the *postset* of t. We will assume that for every transition t, ${}^\bullet t \neq \emptyset$. The semantics of a Petri net is defined as usual: starting from a marking M, a transition t can be fired if for every $p \in {}^\bullet t, M(p) > 0$. This firing results in new marking M' such that $M'(p) = M(p) - |{}^\bullet t \cap \{p\}| + |t^\bullet \cap \{p\}|$. We denote a discrete move of a Petri net from M to M' by $M \to M'$. *Reach*$(\mathcal{U}, M_0)$ denotes the set of reachable markings, that can be obtained via an arbitrary number of moves starting from M_0. Let x be any node of a net. The *cluster* of x (denoted by $[x]$) is a minimal set of nodes of N satisfying following conditions: (i)

$x \in [x]$, (ii) if a place p is in $[x]$, then $p^\bullet \subseteq [x]$, and (ii) if a transition t is in $[x]$, then ${}^\bullet t \subseteq [x]$.

Definition 1: A time Petri net (TPN) $\mathcal{N}$ is a tuple $(\mathcal{U}, M_0, I)$ where $\mathcal{U} = (P, T, F)$ is the underlying net, M_0 is an initial marking, and $I : T \to \mathcal{I}(\mathbb{Q}_{\geq 0})$ associates with each transition a firing interval.

We denote by $eft(t)$ and $lft(t)$ the lower and upper bound of interval $I(t)$. For a TPN $\mathcal{N} = (\mathcal{U}, M_0, I)$ let *Untime*$(\mathcal{N}) = (\mathcal{U}, M_0)$ denote the underlying net i.e., without timing constraints on transitions.

Let us now describe the semantics of Time Petri nets with multi-enabledness. In a multi-enabling semantics, when a marking associates to each place in the preset of a transition a number of tokens that is several times the number of tokens needed for the transition to fire, then this transition is considered as enabled several times, i.e. several occurrences of this transition are waiting to fire. Defining a multi-enabling semantics needs to address carefully which instances of enabled transitions can fire, what happens when an instance of a transition fires, in terms of disabling and creation of other instances. Several policies are possible as discussed in [7], [8], [10]. In the rest of the paper, we adopt a semantics where the oldest instances of transitions fire first, are subject to urgency requirements, and the oldest instances are also the disabled ones when a conflict occurs. We formalize this below.

Definition 2: Let M be a marking of a TPN $\mathcal{N} = (\mathcal{U}, M_0, I)$. A transition $t \in T$ is *k-enabled* at marking M, for $k > 0$, if for all $p \in {}^\bullet t, M(p) \geq k$ and there exists a place $p \in {}^\bullet t$ such that $M(p) = k$. In this case, k is called the *enabling degree* of t at marking M, denoted $deg(M, t)$.

Therefore, from a marking M, and for each transition t, there are exactly $deg(M, t)$ instances of transition t which are enabled. A configuration of a time Petri net associates a clock with each instance of a transition that has been enabled. This is called *threshold semantics*. Time can elapse if no urgency is violated, and an occurrence of a transition t is allowed to fire if it has been enabled for a duration $d \in I(t)$.

Formally, a *configuration* is a pair $C = (M, enab)$, where M is a marking, and $enab$ is a partial function $enab : T \to (\mathbb{R}_{\geq 0})^{\mathbb{N}}$. We denote by $enab(t)_i$ the i^{th} entry of $enab(t)$. For a marking M, $t \in T$, $enab(t)$ is defined iff there exists $k > 0$ such that t is k-enabled at M. We further require that the length of vector $enab(t)$ is exactly $deg(M, t)$, and if $1 \leq i < j \leq deg(M, t)$, then $enab(t)_i \geq enab(t)_j$.

In a configuration $C = (M, enab)$ each enabled transition t is associated with a vector $enab(t)$ of decreasing real values, that record the time elapsed since each instance of t was newly enabled. With this convention, the first enabled instance of a transition t (that must have the maximal clock value) is the first entry $enab(t)_1$ of the vector. For a value $\delta \in \mathbb{R}$, we will denote by $enab(t)+\delta$ the function such that associates $enab(t)_i+\delta$ to the i^{th} occurrence of t. The initial configuration of a TPN is $C_0 = (M_0, v_0(X_0))$, where X_0 is a set of vectors containing $deg(M_0, t)$ clocks per transition t.

A *timed move* from a configuration $C = (M, enab)$ consists in letting δ time units elapse, i.e. move to configuration $C' = (M, enab+\delta)$. Such move is allowed only if it does not violate urgency, i.e. $enab(t)_i + \delta \leq lft(t)$ for every instance i of t.

A *discrete move* consists in firing an instance of a transition t which clock lays in interval $I(t)$. When firing transitions, we will use the *first enabled first fired (FEFF)* policy, that is the instance of transition t fired is the one with the highest value of time elapsed, i.e., $enab(t)_1$. Similarly, we disable transitions according to the *first enabled first disabled (FEFD)* policy. A standard way to address time in TPNs semantics is to start measuring time for a transition as soon as this transition is newly enabled. However, as highlighted in [19], there are several possible interpretations of new enabledness. A frequently used semantics is the *intermediate semantics*, which considers intermediate markings, i.e. when firing a transition t, we consider the marking M'' obtained after removing the tokens in ${}^\bullet t$ from M, and comparing the set of enabled transitions in M'' with those enabled after production of tokens in the places of $t^\bullet$. Let transition t be k-enabled at marking M for $k > 0$, and $enab(t)_1 \in I(t)$. Then, an instance of t can fire, and we obtain a new marking M' via an intermediate marking M'' as follows:

$M''(p) = M(p) - 1$ if p in ${}^\bullet t$ else $M''(p) = M(p)$.

Then we obtain marking M' as :

$M'(p) = M''(p) + 1$ if p in $t^\bullet$ else $M'(p) = M''(p)$.

Firing an instance of a transition t changes the enabling degree several transitions (and not only of t). We will say that a transition is *newly enabled* by firing of t from M iff its enabling degree is larger in M' than in M''. Then newly enabled transitions are attached an additional clock initially set to 0 in $enab$. For transitions which enabling degree decreases (i.e. that are in conflict with t), the first clock in $enab$ is discarded. We refer readers to [4] for a more formal presentation of the semantics. We will write $C \longrightarrow_{\mathcal{N}} C'$ when there exists a move from C to C' allowed by net $\mathcal{N}$.

A *timed firing sequence* of $\mathcal{N}$ starting from configuration $q_1 = (M_1, enab_1)$ is a sequence of timed and discrete moves $\rho = q_1 \xrightarrow{\alpha_1} q_2 \xrightarrow{\alpha_2} \dots \xrightarrow{\alpha_{n-1}} q_n$ where α_i is either a transition of T or a value from $\mathbb{R}_{\geq 0}$. A configuration C is reachable iff there exists a firing sequence from the initial configuration to C. Let *Reach*$(\mathcal{N}, C_0)$ denote the set of reachable configurations of a TPN $\mathcal{N}$, starting from configuration C_0. For a configuration $C = (M, enab)$ we denote by *Untime*$(C) = M$ the *untiming* of configuration C. Similarly, *Untime*(*Reach*$(\mathcal{N})$) denotes the set of markings M such that there exists a reachable configuration (M, v) of $\mathcal{N}$. Note that *Reach* and *Untime* operations are not commutative and *Reach*(*Untime*$(\mathcal{N})$) could be different from *Untime*(*Reach*$(\mathcal{N})$) for a TPN $\mathcal{N}$. A TPN $\mathcal{N}$ is *bounded* if *Untime*(*Reach*$(\mathcal{N})$) is finite.

A *timed word* over T is a word of $(T \times \mathbb{R}_{\geq 0})$. Let $\rho = q_1 \xrightarrow{\alpha_1} q_2 \xrightarrow{\alpha_2} \dots \xrightarrow{\alpha_{n-1}} q_n$ be a firing sequence starting from q_1, and let $i_1, \dots i_k$ denote the indexes of discrete moves in ρ. The timed word associated with ρ is the word $w = (t_1, d_1) \dots (t_k, d_k)$, where each t_j is transition α_{i_j}, $d_1 = \sum_{j<i_1} \alpha_j$, and for every $m > 1$, $d_m = d_{m-1} + \sum_{i_{m-1}<j<i_m} \alpha_j$. A timed word w is *enabled* at configuration C if there exists a timed firing sequence ρ starting from C,

and w is the timed word associated with this sequence. The untiming of a timed word $w = (t_1, d_1) \dots (t_n, d_n)$ is the word $\mathit{Untime}(w) = t_1 \dots t_n$.

For a TPN $\mathcal{N} = (\mathcal{U}, M_0, I)$, let $\mathit{Lang}(\mathcal{N})$ denote the set of all timed words enabled at initial configuration C_0, and $\mathit{ULang}(\mathcal{N})$ be defined as $\mathit{Untime}(\mathit{Lang}(\mathcal{N}))$, i.e., the set of words obtained by removing the timing component from $\mathit{Lang}(\mathcal{N})$. Observe that $\mathit{ULang}(\mathcal{N}))$ can be different from $\mathit{Lang}(\mathit{Untime}(\mathcal{N}))$. Similarly, we let $\mathcal{R}(\mathcal{N}, C_0) = \mathit{Untime}(\mathit{Reach}(\mathcal{N}, C_0))$ be the set of (untimed) reachable markings, i.e., $\mathcal{R}(\mathcal{N}, C_0) = \{M \mid \exists (M, v) \in \mathit{Reach}(\mathcal{N}, C_0)\}$.

A *forced* 0*-delay timed firing sequence* of N is a sequence from $(Q \times T)^\infty$ such that $q_{i-1} \xrightarrow{\alpha_i} q_i$ where α_i is a transition of T with $eft(\alpha_i) = lft(\alpha_i) = 0$. Hence transitions are enabled at each configuration and fired immediately without letting any other transition of the net fire. Specifications with such sequences should be considered as ill-formed, as they contain *mandatory* instantaneous executions of unbounded sets of actions. Note that forbidding such sequences still allows the net to exhibit some sequences that *may* execute in 0 time. For example, a TPN in which each transition has $eft(t) = 0$ and $lft(t) > 0$ can exhibit an infinite sequence of 0-duration, but it has no forced 0-delay timed firing sequence.

III. Processes of untimed and timed nets

We now define a partial-order semantics for timed nets with multi-enabledness using processes. These notions will be central to reason about TPNs and their properties. The notion of time causal process for TPNs has been introduced by [5], and later used by [11] to study a truly concurrent semantics for TPNs. First, we recall the definitions in the untimed setting.

Definition 3 (causal net): A causal net $ON = (B, E, G)$ is a finitary (each node has finite number of predecessors) acyclic net with B as a set of conditions, E as set of events, flow relation $G \subseteq B \times E \cup E \times B$, such that $E = \{e \mid (e, b) \in G\} \cup \{e \mid (b, e) \in G\}$, and for any condition b of B we have $|\{e \mid (e, b) \in G\}| \leq 1$ and $|\{e \mid (b, e) \in G\}| \leq 1$.

When causal nets are used to give a partial order semantics to Petri nets, events represent occurrences of transitions firings, and conditions occurrences of places getting filled with tokens. The last condition in the definition states that a token in a place is produced by a single transition firing, and is consumed by a single transition. As conditions have a single successor, causal nets are *conflict free*. They are hence well suited to represent executions of nets. Let $\prec = G^+$ and $\preceq = G^*$. We define $e{\downarrow}$ as the downward closure $e{\downarrow} = \{f \in E \mid f \preceq e\}$. A set $E' \subseteq E$ is downward closed iff $E'{\downarrow} = E'$.

We define $ON^\bullet = \{b \in B \mid b^\bullet = \emptyset\}$ and ${}^\bullet ON = \{b \in B \mid {}^\bullet b = \emptyset\}$; Note that if we are looking at finite causal nets then $ON^\bullet$ is always defined.

Definition 4 (causal process): Given a PN $\mathcal{U}$, a causal process of $\mathcal{U}$ is a pair $U = (ON, \pi)$ where, π is mapping from $B \cup E \to P \cup T$, satisfying following conditions:

1) $\pi(B) \subseteq P$ and $\pi(E) \subseteq T$
2) $\pi{\downarrow_{e^\bullet}}$ is a bijection between sets $e^\bullet$ and $\pi(e)^\bullet$.
3) $\pi{\downarrow_{{}^\bullet e}}$ is a bijection between sets ${}^\bullet e$ and ${}^\bullet \pi(e)$.
4) For each p, $M_0(p) = |\{b \in {}^\bullet ON \mid \pi(b) = p\}|$.

To relate transitions of a PN and events of a causal net, we will consider that events are of the form $e = (X, t)$, where $\pi(e) = t$ and $X = {}^\bullet e$ i.e., event e corresponds to firing of transition t and X is a set of conditions consumed when firing t. We will also let conditions be of the form $b = (e, p)$, where e is the event that creates condition b, and $p = \pi(b)$ is the place that is represented by condition b. We will write $place(b)$ to denote such place in the original net. We let $posb(ON)$ denote the set of possible events which could be added to ON and is defined as $posb(ON) = \{e = (X, t) \mid X \subseteq B \wedge \pi(X) = {}^\bullet t \wedge \forall x \in X,\ x^\bullet = \emptyset\}$. Following these definitions, one can inductively build processes to capture the semantics of (even unbounded) Petri nets.

Proposition 1: Let $\mathcal{U}$ be any untimed net. For any word $w \in \mathit{Lang}(\mathcal{U}, M_0)$, there exists a causal process U of $\mathcal{U}$, such that $w \in \mathit{Lang}(U)$.

Now we define the notions of time causal net, and time causal process for Time Petri nets with muti-enabling semantics. Similar notions appear in [5] for Time Petri nets.

Definition 5 (time causal net): A time causal net is a tuple $ON = (B, E, G, \tau)$ where (B, E, G) is a causal net, and $\tau : E \to \mathbb{R}_{\geq 0}$ is a timing function s.t. $eG^+e' \Rightarrow \tau(e) \leq \tau(e')$.

For a time causal net $ON = (B, E, G, \tau)$, we define $\mathit{Untime}(ON)$ as the net (B, E, G) i.e., ON without its timing function. Now, given two untimed causal nets $ON = (B, E, G)$ and $ON' = (B', E', G')$, ON' is said to be prefix of ON, denoted by $ON' \leq ON$ if $B' \subseteq B, E' \subseteq E, G' = G \cap (B' \times E' \cup E' \times B')$, where E' is finite and downward closed subset of E. If ON and ON' are time causal nets, E' is required to be timely sound, i.e., for all e' in E' we have $\tau'(e') \leq \tau(e')$.

Definition 6 (time causal process): A time causal process a TPN $\mathcal{N}$ is a pair $N = (ON, \pi)$ where ON is a time causal net, and π is mapping from $B \cup E \to P \cup T$, satisfying conditions 1-4 of a causal process, and:

5) for every event $e = (X, t)$, $\min_{x \in X}\{\tau(e) - \tau({}^\bullet x)\} \in I(\pi(e))$, i.e. the time elapsed between enabling of the occurrence of t represented by e and its firing belongs to $I(t)$.
6) if there exists $X \subseteq ON^\bullet$ and a transition t such that $Place(X) = {}^\bullet t$ and $\tau({}^\bullet x)$ is minimal for every occurrence of a place $place(x)$ in $ON^\bullet$, then $\max(\tau(E_n)) - \max(\tau({}^\bullet X)) < lft(t)$. This last condition means that if a transition was urgent before the date of the last event in ON, then it belongs to the time causal net, or was not appended due to a conflict.

For a time causal process (ON, π) we define $\mathit{Untime}(ON, \pi)$ as $(\mathit{Untime}(ON), \pi)$. As for Petri Nets, for a time causal net $ON = (B, E, G, \tau)$, we denote by $posb(ON)$ (and we define similarly) the set of events that can be appended to ON (regardless of timing considerations). Abusing our notation, for a condition $b = (e, p)$ we will define $\tau(b)$ as $\tau(b) = \tau(e)$.

As for Petri nets, we can show that time causal processes faithfully describe the semantics of Time Petri nets. Given a

time causal process $N = (ON, \pi)$, where $ON = (B, E, G, \tau)$, a timed word of N is a timed word $w = (e_1, d_1) \dots e_{|E|}, d_{|E|}$ such that $e_1 \dots e_{|E|}$ is a linearization of *Untime*(ON), $d_i = \tau(e_i)$. Note that as w is a timed word, this means in addition that for every $i < j$, we have $d_i < d_j$. We denote by *Lang*(N) the set of timed words of time causal process N. Note that there exist some words $w \in$ *Lang*(*Untime*(N)) such that w is not the untiming of a word in *Lang*(N). We can now prove the following proposition:

Proposition 2: Let $\mathcal{P}$ be the set of time causal processes of a time Petri net $\mathcal{N}$. Then, *Lang*$(\mathcal{N}) = \bigcup_{N \in \mathcal{P}}$ *Lang*(N)

Example 1: Consider the FC-TPN (N, M_0) shown in Figure 6. Nets ON_1 and ON_2 in Figure 1 are causal nets of *Untime*(N, M_0). One can notice that $ON_1 \leq ON_2$.

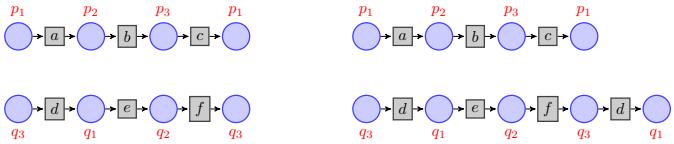

Fig. 1. Untimed causal nets ON_1 and ON_2

Nets ON_3 and ON_4 in Figure 2 below are time causal nets of (N, M_0). We have $ON_3 \leq ON_4$. Let us now compare ON_3 and ON_4 with (untimed) causal processes: we have $ON_1 \leq$ *Untime*(ON_3) and $ON_2 \leq$ *Untime*(ON_4).

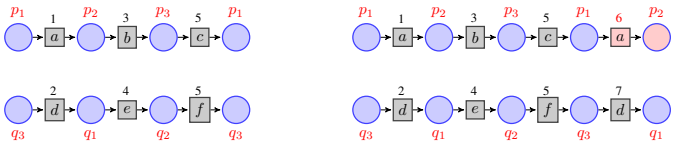

Fig. 2. Time causal net ON_3 and ON_4

IV. Free Choice Time Petri Nets

Time Petri nets with urgent semantics [16] are very expressive. They can be used to model distributed timed systems with unbounded resources. Unsurprisingly, most problems, particularly those listed below are undecidable for this model:
Fireability: Given a TPN $\mathcal{N} = (\mathcal{U}, M_0, I)$ and a transition t, is there a configuration $(M, enab) \in$ *Reach*$(\mathcal{N}, C_0)$ such that t is fireable at $(M, enab)$.
Termination: Given a TPN $\mathcal{N} = (\mathcal{U}, M_0, I)$, does it have a non-terminating run?
Coverability: Given a TPN $\mathcal{N} = (\mathcal{U}, M_0, I)$ and marking M, is there a marking $M' \in$ *Reach*$(\mathcal{N}, M_0)$ such that $M' \geq M$?
Reachability: Given a TPN $\mathcal{N} = (\mathcal{U}, M_0, I)$ and a marking M, decide if $M \in$ *Reach*$(\mathcal{N}, M_0)$.
Boundedness: Given a PN $\mathcal{N} = (\mathcal{U}, M_0, I)$ decide if *Reach*$(\mathcal{N}, M_0)$ is finite.

To obtain decidability, one often considers bounded TPNs, where the number of tokens in places cannot grow arbitrarily. In this case, one can compute a finite *state class graph* [6] representing all runs of the the TPN, on which reachability, coverability, and all regular properties are decidable. However, bounded TPNs cannot represent systems with unbounded resources. Furthermore, it is undecidable in general whether a TPN is bounded. One usually has to rely on a priori known bounds on place contents, or restrict to the class of nets such that *Untime*$(\mathcal{N})$ is bounded.

In this paper, we consider a different structural restriction of TPNs, which is based on the untimed underlying PN, namely free choice. This is a standard restriction in the untimed setting, that allows for efficient algorithms (see [13]). In this section, we show the interesting properties it engenders in TPNs. In section V we show how it affects their robustness under guard enlargement.

Definition 7 (Free choice PN and free choice TPN): A Petri net $\mathcal{U} = (P, T, F)$ is called *(extended) free choice, denoted FC-PN*, if for any pair of transitions t and t' in T: ${}^\bullet t \cap {}^\bullet t' \neq \emptyset \implies {}^\bullet t = {}^\bullet t'$. A TPN $\mathcal{N} = (\mathcal{U}, M_0, I)$ is called a *free choice TPN* (FC-TPN for short), if its underlying untimed net *Untime*$(\mathcal{N}) = \mathcal{U}$ is free choice.

A. Pruning a TPN while preserving reachability

As mentioned above, the fireability and termination problems are undecidable for TPNs in general. In the next section we show that they are decidable for FC-TPNs. As a first step, given an FC-TPN $\mathcal{N}$, whose underlying net is free choice, we construct another FC-TPN $Prune(\mathcal{N})$ in which we remove transitions from a cluster if they can never be fired (due to the lower bound of their time constraint) and tighten timing constraints. Note that we do not delete *all* dead transitions from the net, but remove only transitions for which we can decide locally, by considering their clusters, that they will never fire. Let us illustrate this with an example.

Example 2: Consider the FC-TPN $\mathcal{N}$ in Figure 3. Consider transition b, and its cluster $[b]$. One can notice from the definition of FC-TPNs that all transitions from the same cluster have a new instance created as soon as any transition from the same cluster has a new instance. Note also that in this example, it is clear that transition c will never be fired: in a configuration $(M, enab)$, every instance of c is created at the same time as another instance of b and d, and hence we have $enab(c) = enab(b) = enab(d)$. Let $enab(c) = enab(b) = r_1 \dots r_k$. Transition b has to fire or be disabled at $lft(b) - r_1$, $lft(b) - r_2$, ... If b fires or is disabled, then the oldest instance of c is also disabled. As we have $lft(c) > lft(b)$ every i^{th} instance of b will fire or be disabled before the i^{th} instance of c. Hence one can safely remove transition c without changing the semantics of the net.

Similarly, in the cluster $[e]$, we cannot say for sure that some transition will be never fired, but only that the maximum amount of time any transition can elapse is 2 time units. Note that, in fact, neither transition f nor e is fireable in this net, but we cannot decide it just by looking at the clusters. Indeed in order to decide if e and f are fireable, we have to study the behaviour of the net. Hence our pruning operation does not modify this. Thus, after removing transition c from its cluster, modifying the flow relation appropriately and changing the upper bounds of remaining transitions, we obtain the free choice net in Figure 4. One can see that

$\mathit{Reach}(\mathcal{N}, M_0) = \mathit{Reach}(\mathit{Prune}(\mathcal{N}), M_0)$. Therefore, we also obtain $\mathit{Lang}(N, M_0) = \mathit{Lang}(\mathit{Prune}(N), M_0)$.

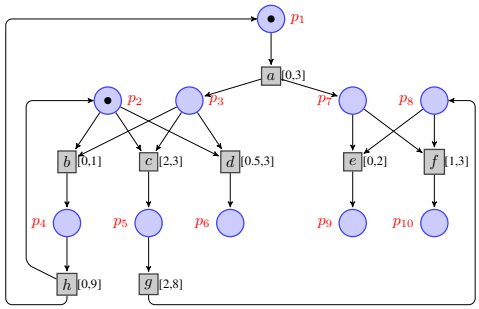

Fig. 3. a FC-TPN $\mathcal{N}$

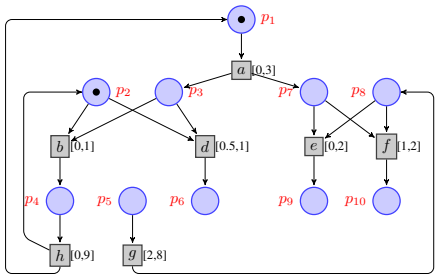

Fig. 4. The pruned net $\mathit{Prune}(\mathcal{N})$

Formally, we can define pruning as follows:

Definition 8 (pruned cluster and pruned net): Given an FC-TPN $\mathcal{N} = (\mathcal{U} = (P, T, F), M_0, I))$, we define the *pruned cluster* of a transition t as $\mathit{Prune}([t]) = \{t' \in [t] \mid eft(t') \leq \min_{t'' \in [t]}(lft(t''))\}$. The pruning of a FC-TPN $\mathcal{N}$ is the *pruned net* $\mathit{Prune}(\mathcal{N}) = (\mathcal{U}' = (P, T', F'), M_0, I')$, where:

- $T' = T{\downarrow}_{\cup_{t \in T} Prune([t])}$.
- $F' = F{\downarrow}_{T'} \subseteq (P \times T') \cup (T' \times P)$.
- For each transition t in T', we define $I'(t) = \big(eft(t), \min_{t' \in [t]} lft(t')\big)$.

Lemma 1 below shows that pruning away unfireable transitions from clusters of a FC-TPN, does not modify its semantics. More precisely, the LTS obtained for a pruned FC-TPN is isomorphic to the LTS for the original net. This is not surprising and has already been considered in [12], where pruning is called 'normalization' and is used to reason about free choice (but not to remove transitions nor places).

Lemma 1 (Pruning Lemma): Given an FC-TPN $\mathcal{N}$, the net $\mathit{Prune}(\mathcal{N})$ is such that $(\mathcal{C}_{\mathcal{N}}, \longrightarrow_{\mathcal{N}})$ is isomorphic to $(\mathcal{C}_{\mathit{Prune}(\mathcal{N})}, \longrightarrow_{\mathit{Prune}(\mathcal{N})})$.

Proof: (Sketch) Let $\mathcal{N}$ be a FC-TPN, and let $\mathcal{N}' = \mathit{Prune}(\mathcal{N})$. We define a relation $\mathcal{R}$ from configurations of $\mathcal{N}$ to configurations of $\mathcal{N}'$, by simply erasing clock values attached to pruned transitions. We can then prove that $\mathcal{R}$ is an isomorphism by induction on the length of runs. Details can be found in [4].

An immediate consequence of lemma 1 is that $\mathit{Lang}(\mathcal{N}) = \mathit{Lang}(\mathit{Prune}(\mathcal{N}))$. Note that this lemma holds only under the free choice assumption. Indeed, in a standard TPN, one can disable the most urgent transition from a cluster without disabling other instances of transitions in this cluster. In the example of Figure 5, for instance, transition t_2 and t_3 belong to the same cluster, and pruning would remove t_3. However, in the unpruned net, transition t_2 can be disabled by firing of t_1, which still allows t_3 to fire later. Hence, for this non-FC-TPN, $\mathit{Lang}(\mathcal{N}) \neq \mathit{Lang}(\mathit{Prune}(\mathcal{N}))$.

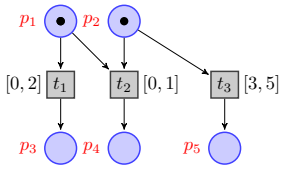

Fig. 5. Pruning only works for FC-TPNs

B. Simulating runs in the timed and untimed FC-TPN

We now prove our main theorem, which relates time causal processes of pruned FC-TPNs with untimed causal processes of their untimed nets.

Theorem 1 (Inclusion of untimed prefixes): Let $\mathcal{N} = (\mathcal{U}, M_0, I)$ be a pruned FC-TPN (without forced 0-delay time firing sequences) and let $\mathcal{U}' = \mathit{Untime}(\mathcal{N})$. Let U' be an (untimed) causal process of $\mathcal{U}'$. Then there exists a time causal process N of $\mathcal{N}$ such that $U' \leq \mathit{Untime}(N)$.

Proof: We are given U' a causal process of $\mathcal{U}'$. We will iteratively construct a pair ρ_i, σ_i, where ρ_i is a causal process of $\mathcal{U}'$, σ_i a time causal process of $\mathcal{N}$, and such that ρ_i is a prefix of $\mathit{Untime}(\sigma_i)$. The construction ends at some $n \in \mathbb{N}$ such that $\rho_n = U'$. At that stage of the algorithm, σ_n is a time causal process such that $U' \leq \mathit{Untime}(\sigma_n)$ which is the desired result. For this, we maintain the following invariants at every intermediate step, i.e., for all $0 \leq i \leq n$:

(I1) ρ_i is a prefix of U' and
(I2) ρ_i is a prefix of $\mathit{Untime}(\sigma_i)$
(I3) either $|\rho_i| + 1 = |\rho_{i+1}|$ or $|\sigma_i| + 1 = |\sigma_{i+1}|$
(I4) $e \in posb(\rho_i)$ and $e \notin \sigma_i$ implies that $e \in posb(\sigma_i)$
(I5) $eG_i^+e' \Rightarrow \tau(e) \leq \tau(e')$, (with G_i^+ the flow relation of σ_i).

The first two conditions have been explained above. Condition $(I3)$ ensures that the algorithm progresses at every iteration, either in ρ_i or σ_i. Condition $(I4)$ says that if an event e is enabled in the untimed causal process ρ_i and has not yet been fired in σ_i, then it must be enabled at σ_i. Note that due to urgency, e may still be not fireable in σ_i. Finally, Condition $(I5)$ ensures that the time stamps that we add in σ_i are consistent with its causal order.

We start by defining, for a time causal process ON, the maximal firing date for an event $e = (X, t) \in posb(ON)$ as $mfd(ON, e) = \max_{x \in X}(\tau(x)) + lft(t)$. This represents the maximal date that can be attached to event $e = (X, t)$ when it is appended to ON. Further, we use $td(ON, e)$ to denote the difference between $mfd(ON, e)$ and the maximal date attached to an event in ON. Note that this value can be negative if the maximal date in ON is above $mfd(ON, e)$, i.e., e has already been fired in ON. This represents the time that has to elapse before (or has elapsed after) event e corresponding to firing of a transition t becomes urgent.

Finally, for a (time) process ON and an event e, we use $ON \cup \{e\}$ to denote the (time) process obtained by appending event e to ON, when it is possible to do so. Now we start from an untimed causal process and describe a time-stamping algorithm to obtain a time causal process in Algorithm 1.

Algorithm 1: Causal-net-to-time-causal-net

Input: $\mathcal{N} = (\mathcal{U}, M_0, I)$ a pruned FC-TPN without forced 0-delay firing sequences, $U' = (B', E', G')$ a (untimed) causal process of $\mathcal{U}' = \mathit{Untime}(\mathcal{N})$

Output: N a time causal process of $\mathcal{N}$ such that $U' \leq \mathit{Untime}(N)$

1 Initializations: // ρ_i (resp. σ_i) is a untimed (resp. time) causal process
2 $CLK := 0, \rho_0 := (B_0, \emptyset, \emptyset), \sigma_0 := (B_0, \emptyset, \emptyset, \emptyset)$;
3 **while** $\rho_i \neq U'$ **do**
4 Choose an event $e = (X_e, t_e)$ from $posb(\rho_i) \cap U'$;
5 **if** $e \in \sigma_i$ **then**
6 $\rho_{i+1} \leftarrow \rho_i \cup \{e\}$; $\sigma_{i+1} \leftarrow \sigma_i$;
7 **else**
8 $min = \min_{e' \in posb(\sigma_i)} mfd(\sigma_i, e')$;
9 $S_i = \{e' \in posb(\sigma_i) \mid mfd(\sigma_i, e') = min$ and $min \neq \infty\}$;
10 **if** $S' = S_i \cap E' \neq \emptyset$ **then**
11 Pick an event $e_t = (X, t)$ from S';
12 **else**
13 **if** $S_i = \emptyset$ **then**
14 $CLK' = CLK + max\{0, td(\sigma_i, e)\}$;
15 $\tau(e) = CLK'$;
16 $\sigma_{i+1} \leftarrow \sigma_i \cup \{e\}$; $\rho_{i+1} \leftarrow \rho_i$;
17 $GOTO$ Step-23;
18 **else**
19 Pick an event $e_t = (X, t)$ from S_i;
20 $CLK' = CLK + lft(t) - td(\sigma_i, e_t)$;
21 $\tau(e_t) = CLK'$ // adding time stamp
22 $\sigma_{i+1} \leftarrow \sigma_i \cup \{e_t\}$; $\rho_{i+1} \leftarrow \rho_i$;
23 $i \leftarrow i + 1$;

Let us now explain the algorithm. At initialization, B_0 is the set of conditions corresponding to marked places in M_0, and CLK, a real valued variable which stores the time-elapsed till now, is set to 0. ρ_i and σ_i are respectively, untimed and time causal processes at i-th iteration. S_i is the set of events that are the most urgent instances of transitions in σ_i. The idea is that, at each iteration, we pick (in line 4) an event e enabled in the current ρ_i, which would grow ρ_i to eventually reach the untimed causal process U'. If this event has already been fired in σ_i, then we just add it to ρ_i and go to the next iteration.

Else, we try to fire it in σ_i. However, to do so, we first compute S_i the set of all urgent transitions in σ_i (line 8–9). If there is an urgent transition instance e_t whose corresponding event is also in U', then we pick it (in line 11) and fire it, i.e., add it to σ_i and update time information correctly (line 20-23). This makes sure σ_i has grown at this iteration so we go the next iteration. On the other hand, if there is no urgent transition in S_i which is also in U', we check if there is no urgent transition at all, i.e., S_i is empty. In this case, we elapse time till e can fire and fire it as soon as possible (line 14), updating clocks appropriately.

Finally, if there is some urgent transition in S_i that is not in U', then we fire it as late as possible (line 20–23). The fact that this does not change the enabling of e (due to conflicts) is proved by our invariant $I4$.

Now, we conclude the proof of Theorem 1 from the two following lemmas, whose complete proofs are provided in [4].

Lemma 2: Algorithm 1 preserves all invariants at each iteration of the while loop.

If invariants $I1, I2, I3, I4$, and $I5$ are satisfied for all iterations then Algorithm 1 is correct, i.e. when it stops it has built a time causal process N of $\mathcal{N}$ such that $U \leq Untime(N)$. It remains to show that the algorithm always terminates.

Lemma 3 (Termination Lemma): Algorithm 1 terminates in finitely many steps.

Proof: (*sketch)* We show that σ cannot grow unboundedly. Each step of the algorithm adds either an event to ρ_i, or to σ_i. While the number of steps that add events to ρ_i is finite, the algorithm may still not terminate if σ can grow unboundedly, i.e. the while loop is exited at line 22 unboundedly without progress in ρ. To show that this is not possible, at every step, a set of events from U' are listed as possible events. Now, the crucial idea is that since there are no forced 0-delay firing sequences, time can and must progress, so these events cannot be ignored forever. Hence, σ_i can grow only up to some finite point.

We can now extend Theorem 1 from causal nets to words. Let $\mathcal{U} = (P, T, F)$ be an untimed Petri net. Given a causal process $U = (ON = (B, E, G), \pi)$ of $\mathcal{U}$, consider the partial order $\preceq = G^*$. We denote by $lin(U)$, the set of words over alphabet of transitions obtained by considering linearizations of the partial order of G^* and projecting onto the labeling alphabet (of transitions T), i.e., $lin(N) = \{\rho \in T^* \mid$ there exists ρ' a linearization of G^* such that $\pi|_E(\rho') = \rho\}$.

Now, recall that given an untimed net $\mathcal{U}$, its language $Lang(\mathcal{U}, M_0)$ is a set of firing sequences, i.e., words over the alphabet of transitions. Now, we obtain our characterization for words rather than causal processes.

Corollary 1 (Words version): Let $\mathcal{N}$ be a pruned FC-TPN (without forced 0-delay time firing sequences) and let $\mathcal{U}' = \mathit{Untime}(\mathcal{N})$. Then, for each $w \in Lang(\mathcal{U}', M_0)$ there exists a time causal process N of $\mathcal{N}$ and $w' \in \mathit{Lang}(\mathit{Untime}(N))$ such that $w \preceq w'$.

Proof: Given a word $w \in Lang(\mathcal{U}', M_0)$, using Proposition 1, we get an untimed causal process U' of net $\mathcal{U}'$, such that $w \in lin(U')$. By Theorem 1, we get a time causal process N of net $\mathcal{N}$ such that $U' \leq \mathit{Untime}(N)$. Now since $w \in lin(U')$, and $U' \leq \mathit{Untime}(N)$, we can extend w to $w' \in lin(\mathit{Untime}(N))$.

V. Decidability results for FC-TPNs

In this section, we will show some consequences of the above characterization.

Theorem 2 (Fireability): Given an FC-TPN $\mathcal{N} = (\mathcal{U}, M_0, I)$ (without forced 0-delay time firing sequences), and a transition $t \in T$, checking fireability of transition t in $\mathcal{N}$ is decidable.

Proof: Given a FC-TPN $\mathcal{N}$, one can compute an equivalent pruned version $\mathcal{N}_{Pruned}$, i.e. a Petri nets with the same timed language and the same set of processes, but which clusters have only fireable transitions. One can compute a Petri net $\mathcal{U}' = \mathit{Untime}(\mathcal{N}_{Pruned})$. For every PN, it is well known that coverability of a marking is decidable [18]. A particular transition t is fireable in a Petri net $\mathcal{U}$ iff its preset ${}^\bullet t$ is coverable. Coverability can be obtained by construction of a coverability tree, or using backward algorithms (see for instance [14] for recent algorithms). In both cases, one can exhibit a sequence of transitions witnessing coverability of a particular marking. If $w = t_0 \dots t_k$ is such a sequence witnessing coverability of ${}^\bullet t$ from M_0 in $\mathit{Untime}(\mathcal{N}_{Pruned})$, then one can immediately infer that $w.t \in \mathit{Lang}(\mathcal{U}')$, and hence t is fireable in $\mathcal{U}'$. Using Corollary 1, there exists a timed word $v = (t_0, d_0) \dots (t_k, d_k).(t, d_k + 1) \in \mathit{Lang}(\mathcal{N}_{Pruned})$, and hence $v \in \mathit{Lang}(\mathcal{N})$. Conversely, assume that t is not fireable in $\mathit{Untime}(\mathcal{N})$ and that t it is fireable in $\mathcal{N}$. Then there exists a run ρ of $\mathcal{N}$ firing t. Then, $\mathit{Untime}(\rho)$ is a run of $\mathit{Untime}(\mathcal{N})$ which fires t (contradiction).

Theorem 3 (Termination): Given an FC-TPN $\mathcal{N}$ (without forced 0-delay time firing sequences), it is decidable if $\mathcal{N}$ terminates.

Proof: Let $\mathcal{N}$ be a FC-TPN and let $\mathcal{N}_p$ be its pruned version. Since the reachability graphs of $\mathcal{N}$ and $\mathcal{N}_p$ are isomorphic by Pruning Lemma 1, it is sufficient to decide if $\mathcal{N}_p$ has only terminating runs. Let $\mathcal{N}' = \mathit{Untime}(\mathcal{N}_p)$. If $\mathcal{N}'$ does not terminate, then it allows sequences of transitions that are unboundedly long. As we know from Corollary 1 that for every word w' of $\mathit{Lang}(\mathcal{N}')$, there exists a timed word w of $\mathit{Lang}(\mathcal{N}_p)$ of length $|w| \geq |w'|$, then $\mathcal{N}_p$ (and hence $\mathcal{N}$) allows sequences of transitions of unbounded lengths, i.e., it does not terminate either. In the other direction, if $\mathcal{N}_p$ has an infinite run, as time constraints in free choice TPNs can only prevent occurrence of a transition, then the untimed net clearly has an infinite run too. Thus, we have reduced the problem to termination of an untimed Petri net, which is decidable by the classical coverability tree construction.

The proof technique using relation between untimed processes and processes of untimed nets does not work for all properties, in particular for reachability. Consider, e.g, the net shown in Figure 6. Marking $\{p_1, q_1\}$ is not reachable from the initial marking $\{p_1, q_3\}$ with a multi-server semantics. However, it is reachable in the corresponding untimed net. Note also that marking $\{p_1, q_3\}$ is also reachable for this net under a weak semantics (as proposed in [19]), which proves that strong multi-server and weak semantics of TPNs differ.

VI. Robustness of FC-TPNs to Guard Enlargement

As mentioned in the introduction, robustness addresses preservation of properties of systems that are subject to imprecision of time measurement. In this section, we show

Fig. 6. Pruning and untiming does not preserve reachability

decidability of robustness of fireability for our class of FC-TPNs. Note that this class contains unbounded nets. To the best of our knowledge, this is the first decidability result of robustness for an unbounded class of TPNs allowing urgency. To do this, we use the reduction from fireability checking in FC-TPNs to fireability checking in the untimed underlying net shown in section V.

Let us now formally define the notion of robustness and the problems considered in this paper. Let $int = [a, b]$ be a time interval. The *enlargement* of int by $\Delta > 0$ is the interval $int_\Delta = [\max(0, a - \Delta), b + \Delta]$. Let $\mathcal{N} = (\mathcal{U}, M_0, I)$ be a FC-TPN. The *enlargement* of $\mathcal{N}$ by $\Delta > 0$ is the net $\mathcal{N}_\Delta = (\mathcal{U}, M_0, I_\Delta)$, obtained from $(\mathcal{U}, M_0, I)$ by replacing every interval $I(t)$ by its enlarged version $I(t)_\Delta$. The robustness problem for a chosen enlargement $\Delta > 0$ consists of deciding whether properties of a net (reachability, boundedness, coverability) or its (untimed) language are preserved in $\mathcal{N}_\Delta$. A more general robustness question asks whether there exists a value for Δ such that $\mathcal{N}_\Delta$ preserves reachability, coverability, or the (untimed) language of $\mathcal{N}$. A positive answer to this question would imply that changing guards slightly preserves the considered property, i.e. this property is robust w.r.t a small time imprecision.

In general, robustness problems are undecidable for TPNs, as shown in [3], and become decidable when the considered nets are bounded. An interesting question is whether robustness of some of the above mentioned problems is decidable for TPNs with multiple enabling for unbounded classes of nets. Answering this would provide useful tools to check properties of systems of bounded timed processes communicating through bag channels.

Let us now consider robustness of fireability: let $\mathit{Fireable}(\mathcal{N})$ denote the set of transitions which are fireable in $\mathcal{N}$. We will say that a net $\mathcal{N}$ is robust w.r.t fireabiliy of transition iff there exists a real value $\Delta > 0$ such that $\mathit{Fireable}(\mathcal{N}) = \mathit{Fireable}(\mathcal{N}_\Delta)$. Intuitively, this property says that in an imperfect time measurement setting, there exists some precision of clocks that allows to preserve fireability of transitions of $\mathcal{N}$ during an implementation. It is easy to observe (see [4] for details) that all FC-TPNs are not robust.

Proposition 3: The class of FC-TPNs is not robust wrt fireability of transitions.

As fireability of every transition in a FC-TPN is decidable, for a fixed enlargement Δ, one can decide whether the fireability set of $\mathcal{N}$ differs from that of $\mathcal{N}_\Delta$. So the next question is to decide whether there exists a $\Delta > 0$ such that net $\mathcal{N}$ is robust with respect to fireability.

Definition 9 (Fireability robustness problem): Given a TPN $\mathcal{N}$, does there exist $\Delta \in \mathbb{Q}_{>0}$ such that $\mathit{Fireable}(\mathcal{N}) =$

Fireable($\mathcal{N}_\Delta$)?

Theorem 4: Let $\mathcal{N}$ be a FC-TPN without forced 0-delay time firing sequences. Then robustness of fireability is decidable. If $\mathcal{N}$ has robust fireability, then one can effectively compute a value Δ such that *Fireable*($\mathcal{N}$) = *Fireable*($\mathcal{N}_\Delta$).

Proof: We first observe that in a pruned FC-TPN $\mathcal{N}$, a transition t is fireable iff all transitions in t's cluster are fireable. Next, all timed words of $\mathcal{N}$ are also timed words of $\mathcal{N}_\Delta$. So, enlarging a net can only result in new behaviors. As a consequence, if a transition t is fireable in $\mathcal{N}$, it is necessarily fireable in $\mathcal{N}_\Delta$. Further, note that as they have the same underlying untimed net, $\mathcal{N}$ and $\mathcal{N}_\Delta$ are both free choice and they have the same clusters. Now the pruning operation applied to each of these nets results in removing transitions (it can never add transitions) and pruning lemma holds for both of them. If after the pruning, we still have the same clusters, then the set of fireable transitions remains the same. The only way to have an extra transition that can fire in $Prune(\mathcal{N}_\Delta)$ is if this transition t in cluster C has been removed in $Prune(\mathcal{N})$ but remains in $Prune(\mathcal{N}_\Delta)$ due to enlargement. By our pruning construction there must exist another fireable transition in this cluster in $Prune(\mathcal{N})$ (else we would not remove t):

Claim: For any $\Delta > 0$, *Fireable*($\mathcal{N}$) $\neq$ *Fireable*($\mathcal{N}_\Delta$) iff there exists a cluster C_Δ of $Prune(\mathcal{N}_\Delta)$ such that $C \subset C_\Delta$ for some cluster C of $Prune(\mathcal{N})$, and at least one transition t of C is fireable.

Thus, it suffices to look at each cluster of $Prune(\mathcal{N})$ and compute the smallest possible enlargement that does not give new behaviors. More formally, two intervals I_1, I_2 are *neighbors* if the smallest closed intervals containing them have a non-empty intersection. Given two intervals I_1 with end-points $a \leq b$ and I_2 with end-points $c \leq d$ that are not neighbors and such that $b < c$, then one can easily compute a value $\Delta_{I1,I2} < (c-b)/2$. One can for instance choose $\Delta_{I1,I2} = \frac{(c-b)}{4}$. For a cluster C of $\mathcal{N}$ with transitions $t_1 \dots t_k$ such that $t_1 \dots t_i$ are not pruned, and $t_{i+1} \dots t_k$ are pruned, we have $eft(t_j) > min\{lft(t_q), q \in 1..i\}$ for every $j \in i+1..k$. Similarly we can compute $\Delta_C = min\{eft(t_j) - min\{lft(t_q), q \in 1..i\}\}$. Then, enlarging $\mathcal{N}$ by $\frac{\Delta_C}{4}$ does not change the set of transitions preserved by pruning. Now, if any transition of $t_{i+1} \dots t_k$ is a neighbor of $[0, min\{lft(t_q), q \in 1..i\}]$, such a Δ_C does not exist.

Consequently to check robustness of fireability, it suffices to check existence of a value Δ_C for each cluster C of $\mathcal{N}$ that has a fireable transition. If one such cluster does not allow computing a strictly positive enlargement, then the net is not robust. Otherwise, it suffices to choose as Δ the smallest value allowing enlargement encountered for a cluster of $\mathcal{N}$. Clearly, enlarging $\mathcal{N}$ by Δ does not change the fireability set of $\mathcal{N}$.

From the above proof we can characterize a class of unbounded FC-TPNs for which robustness of fireability set is guaranteed by definition.

Corollary 2: Let $\mathcal{N}$ be a FC-TPN such that *Untime*(*Prune*($\mathcal{N}$)) = *Untime*($\mathcal{N}$). Then the fireablility set of $\mathcal{N}$ is robust w.r.t guard enlargement.

VII. Discussion

The technique used to obtain decidability of fireability and termination holds for free choice TPNs with a multiple server semantics, when we disallow forced 0-delay infinite runs. All these conditions are necessary for our proofs to hold.

Let us first address the free choice assumption, without which the problems considered are undecidable. Indeed, the two counter machine encoding of [15] relies on urgency and uses non-free choice nets, in which transitions have at least one 1-bounded place in their preset. One can easily show that this encoding works even under a multiple server semantics. In particular, Theorem 1 does not hold without the free choice assumption. Consider the non-free choice net $\mathcal{N}_{nfc}$ in Figure 7 (left). A causal process for *Untime*($\mathcal{N}_{nfc}$) is shown in Figure 7(right, top). One can verify in this untimed net that transition c is fireable. However, there is no way to build a time causal net that contains this causal net. Indeed, transition c is not fireable in $\mathcal{N}_{nfc}$ and one cannot add event e_3 in the only time causal net defined by $\mathcal{N}_{nfc}$ depicted in Figure 7 (right, bottom). Note also that marking $\{p_1, p_3\}$ is reachable in *Untime*($\mathcal{N}_{nfc}$) (and allows firing of c), but not in $\mathcal{N}_{nfc}$.

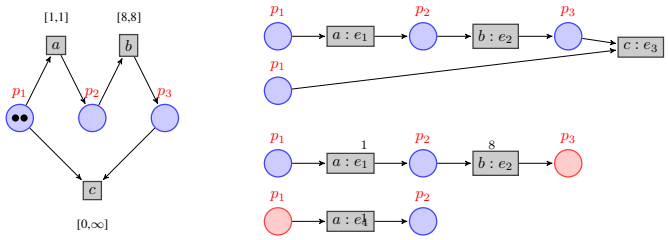

Fig. 7. A non-free choice TPN $\mathcal{N}_{nfc}$ (left), and a causal process (right, up) and a time causal process (right, below) for it.

Next, we discuss the choice of a multi-server semantics. With this semantics, one can consider that a new clock is initialized at any enabling of a cluster (as all transitions from a cluster carry the same set of clocks). Furthermore, when a transition is fired, all remaining instances of transitions in a cluster keep their clocks unchanged. Such a semantics is meaningful and arguably powerful enough, for instance, to address business processes with time that do not contain fork-join constructs, i.e. where case processing are sequences of choices up to completion. Unsurprisingly, with a single server semantics, that keeps only one clock per transition instead of one for every instance of a multi-enabled transition, Lemma 1 does not hold. Indeed, in a single-server semantics, the clock of a fired transition is reset every time the transition fires. In this setting, firing a transition postpones originally expected maximal firing dates of the next occurrences of that transition, which may allow firing of next instances of other transitions that are pruned in our construction. Hence, one cannot easily detect that a transition is useless.

Finally, the 0-delay assumption forbids an arbitrary number of transitions to occur at the same time instant. If this condition is not met, then our algorithm used to build a time process of $\mathcal{N}$ from an untimed process of *Untime*($Prune(\mathcal{N}_P)$) does not

necessarily terminate (Lemma 3). Furthermore, eagerness of urgent transitions firing in zero time can prevent other transitions from firing, which may result in discrepancies between untiming of time processes of a net and untimed processes of the untiming of this net. Consider the pruned and free choice net $\mathcal{N}_2$ depicted in the Figure below. The only allowed executions of $\mathcal{N}_2$ are $\mathit{Lang}(\mathcal{N}_2) = \{(a,0)^k \mid k \in \mathbb{N}\}$. Hence, $\mathcal{N}_2$ has a forced 0-delay firing sequence a^ω, and transition b is not fireable. However, $\mathit{Untime}(\mathcal{N}_2)$ allows sequences of the form $a^k.b$, and hence transition b is fireable. Note that this does not mean that there is no effective procedure to build a time causal net from an untimed causal net, and our algorithm can be adapted to handle Zeno behaviors.

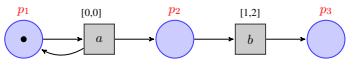

Fig. 8. A free choice TPN $\mathcal{N}_2$ with Zeno behavior

Open issues: We have shown decidability of fireability, termination, and robustness of fireability for FC-TPNs with no forced 0-delay time firing sequences. However, our proof technique does not apply directly for closely-related problems such as coverability and boundedness, that are hence still open.

Consider the FC-TPN $\mathcal{N}_{cov}$ shown in Figure 9. Clearly, when removing time constraints from $\mathcal{N}_{cov}$, the obtained (untimed) Petri net allows sequence of transitions $a.b$, which leaves the net in a marking M such that $M(p_1) = M(p_3) = 1$ and $M(p_2) = 0$. However, the timed language of $\mathcal{N}$ contains only $w = (a,1)(a,2)(b,8)$ and its prefixes. So, marking M is not reachable or even coverable by $\mathcal{N}$. Thus, unlike for fireability and termination, one cannot immediately decide coverability (or reachability) of a marking in an FC-TPN just by looking at its untimed version. This does not contradict Corollary 1: once untimed, the causal process for w does indeed contain the causal process associated with $a.b$.

Next, the question of boundedness also does not immediately follow on the same lines as the proofs of theorems 2 and 3. Consider, for instance, the net $\mathcal{N}_{bd}$ of Figure 10. The untimed version of this net allows sequences of transitions of the form $(t_1.t_2)^k$, for any $k \in \mathbb{N}$. At each iteration of this sequence, place p_3 receives a new token. This net is free choice, and following the result of Theorem 1, for every untimed process N_k associated with a sequence $(t_1.t_2)^k$, there exists a time process N'_k of $\mathcal{N}_{bd}$. However, this process necessarily contains as many occurrences of t_3, t_1 as t_2. As t_3 occurs as soon as p_3 is filled, the only sequences of moves allowed by N_k are of the form $(t_1.t_2.t_3)^k$, and hence $\mathcal{N}_{bd}$ is bounded, even if $\mathit{Untime}(\mathcal{N}_{bd})$ is unbounded.

Despite these remarks, we conjecture that we can indeed modify the techniques in this paper to get decidability for coverability and boundedness problems for FC-TPNs. It is unclear whether reachability for FC-TPNs would similarly be decidable. Finally, several robustness issues are still open. We think the results in this paper can be used as a starting point for future work on yet unsolved robustness questions.

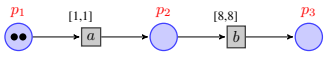

Fig. 9. A free choice TPN $\mathcal{N}_{cov}$

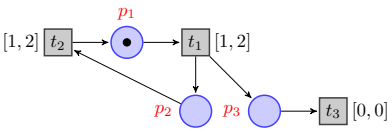

Fig. 10. A net $\mathcal{N}_{bd}$ that is bounded, but whose untimed version is not

ACKNOWLEDGMENT

This work was partly supported by the DST-INSPIRE faculty award [IFA12-MA-17], LIA InForMeL and the Indo-French CEFIPRA project AVERTS.

REFERENCES

[1] P.A. Abdulla and A. Nylén. Timed Petri nets and BQOs. In *Proc. of ICATPN 2001*, volume 2075 of *LNCS*, pages 53–70, 2001.

[2] S. Akshay, B. Genest, and L. Hélouët. Decidable classes of unbounded Petri nets with time and urgency. In *PETRI NETS'16*, volume 9698 of *LNCS*, pages 301–322, 2016.

[3] S. Akshay, L. Hélouët, C. Jard, and P-A Reynier. Robustness of time Petri nets under guard enlargement. *Fundam. Inform.*, 143(3-4):207–234, 2016.

[4] S. Akshay, L. Helouet, and R. Phawade. Combining free choice and time in Petri nets. Technical report, IIT Bombay, 2016. www.cse.iitb.ac.in/internal/techreports/reports/TR-CSE-2016-79.pdf.

[5] T. Aura and J. Lilius. A causal semantics for time Petri nets. *TCS*, 243(1-2):409–447, 2000.

[6] B. Berthomieu and M. Menasche. An enumerative approach for analyzing time Petri nets. In *IFIP Congress*, pages 41–46, 1983.

[7] H. Boucheneb, D. Lime, and O.H. Roux. On multi-enabledness in time Petri nets. In *Proc. of PETRI NETS'13*, volume 7927 of *LNCS*, pages 130–149, 2013.

[8] M. Boyer and M. Diaz. Multiple enabledness of transitions in Petri nets with time. In *Proc. of PNPM'01*, pages 219–228. IEEE, 2001.

[9] D. Bushin and I. Virbitskaite. Time process equivalences for time Petri nets. In *Workshop on Concurrency, Specification and Programming*, volume 1269 of *CEUR Workshop*, pages 257–268, 2014.

[10] A. Cerone and A. Maggiolo-Schettini. Time-based expressivity of time Petri nets for system specification. *TCS*, 216(1-2):1–53, 1999.

[11] T. Chatain and C. Jard. Complete finite prefixes of symbolic unfoldings of safe time Petri nets. In *ICATPN'06*, pages 125–145. 2006.

[12] T. Chatain and C. Jard. Back in time Petri nets. In *Proc. of FORMATS'13*, volume 8053 of *LNCS*, pages 91–105, 2013.

[13] J. Esparza and Jörg Desel. *Free Choice Petri nets*. Cambridge University Press, 1995.

[14] A. Finkel and J. Leroux. Recent and simple algorithms for Petri nets. *Software and System Modeling*, 14(2):719–725, 2015.

[15] N.D. Jones, L.H. Landweber, and Y.E. Lien. Complexity of some problems in Petri nets. *TCS*, 4(3):277–299, 1977.

[16] P.M. Merlin. *A Study of the Recoverability of Computing Systems*. PhD thesis, University of California, Irvine, CA, USA, 1974.

[17] A. Puri. Dynamical properties of timed automata. *In DEDS*, 10(1-2):87–113, 2000.

[18] C. Rackoff. The covering and boundedness problem for vector addition systems. *TCS*, 6:223–231, 1978.

[19] P.A. Reynier and A. Sangnier. Weak time Petri nets strike back! In *CONCUR*, volume 5710 of *LNCS*, pages 557–571. 2009.

[20] V.V Ruiz, D. de Frutos-Escrig, and F. Cuartero. Timed processes of timed Petri nets. In *Proc. of ICATPN'95*, volume 935 of *LNCS*, pages 490–509, 1995.

[21] J. Winkowski. Algebras of processes of timed Petri nets. In *CONCUR '94*, volume 836 of *LNCS*, pages 194–209, 1994.