Time Processes for Time Petri Nets

Tuomas Aura and Johan Lilius

Digital Systems Laboratory
Helsinki University of Technology
FIN-02150 ESPOO
FINLAND

Abstract. Time Petri nets are Petri nets extended with a notion of time, where the occurrence time of a transition is constrained by a static interval. The objective of this work is to give time Petri nets a partial order semantics, based on the nonsequential processes semantics for untimed net systems. A time process of a time Petri net is defined as a traditionally constructed causal process that has a valid timing. This means that the events of the process are labeled with occurrence times which must satisfy specific validness criteria. These criteria are obtained by analyzing how the timing constraints interact with the causal ordering of the events in the net. An efficient algorithm for checking then validness of a given timing is sketched. Interleavings of the time processes are defined as linearizations of the causal partial order of events where also the temporal ordering of events is preserved. The relationship between the firing schedules of a time Petri net and the interleavings of the time processes of the net is shown to be bijective. Also, a sufficient condition is given for when the invalidity of timings for a process can be inferred from an initial subprocess. An alternative characterization for the validness of timings then results in an algorithm for constructing the set of all valid timings for a process. This set of all valid timings is presented as sets of alternative linear constraints, from which the existence of a valid timing can be decided.

`Keywords`: time Petri nets, processes, timing analysis, partial order semantics, causality, net theory

1 Introduction

Petri nets [16,11] are a formalism for modeling and analyzing distributed and concurrent systems. They are characterized by fine grained control over concurrency and synchronization, and equal emphasis of state and actions. Petri nets describe the causal behavior of systems, thus making them a natural candidate for the modeling of distributed systems, because one of the distinctive characteristics of distributed systems is the lack of global time. In practical system, however, timing of events is often just as important as the causal order, as most distributed systems have non-ideal features like timeouts and alarms. Furthermore, performance aspects force designers of distributed systems, such as communication protocols, to maintain synchrony between concurrent subsystems with local clocks. The difficulties in designing time dependent distributed systems are likely to increase the demand for formal methods with timing capabilities. Consequently, time extensions are being planned, for example, to the LOTOS specification language [12].

Also many time related extensions of Petri net formalisms have been introduced to facilitate performance analysis. Some of them attach the timing information on top of the systems, without having any effect on the causal relations between events (eg. [15]). Recently, more attention has been given to net classes where time constraints restrict the causal behavior of the system and limit its state space (eg. [6]). Of these time Petri nets [10] are perhaps the simplest formalisms for modeling systems where time limits can force events to occur and keep others from happening.

In time Petri nets, there is an upper and a lower bound for the time an event can remain enabled without occuring after its preconditions are met. This can be used to model time limits in system specifications and imprecise timing like skew of local clocks, as well as asynchronous timer interrupts. The upper time bound of one potential event can limit the time when another conflicting event can occur, creating dependences not seen in the simple causal view of the system. Furthermore, the timing limits can be used in a way giving the net class the expressive power of a universal computer [13]. This is a strong indication that the analysis on time Petri nets is more complicated than it is for untimed net classes. Even though timing constraints reduce the number of reachable markings of the net they, instead, increase the cost of examining the state space. In addition to the causal state, clock information has to be carried along throughout the analysis. In reachability graphs of time Petri nets [2], the states of the net contain a marking and intervals of possible firing times of all enabled transitions. The states are grouped into state classes where the possible firing times are presented with sets of inequalities. Complex rules are given for transforming the state classes in firings and for checking their equivalence. A more comprehensible reachability graph has been proposed by [14]. In it, a state contains both the marking and the readings of the local clocks of enabled transitions. Thus state changes are divided in two types: either time passes, ie. the clocks are advanced or a transition fires, ie. the marking is changed.

An alternative approach to inspecting the behavior of concurrent systems is to use a partial order semantics, like nonsequential processes [4] and branching processes [7]. The research on branching processes has lately lead to efficient model checking algorithms for net systems [8]. It is rather obvious that the benefits of the partial order approach to the analysis of systems should be assessed also with regard to the analysis of timed systems. Processes have been defined and successfully utilized for some net classes with time [18,9,17], but in these cases the timing does not interfere with the causal order of events. We aim to show that nonsequential processes can be successfully used in presenting the behavior of time Petri nets. Although the occurrence times of events seem to put even causally unrelated events in a sequential order, the concurrent parts of the process develop independently within the specified time limits. In a causal system, the relative speed of concurrent events can be arbitrary and order results only from synchronization by shared events. In time constrained systems, the speeds are given some bounds, but the events are not forced into a fixed sequence. Instead, the order of events arises from synchronizing events, or from the time limits of enabled events representing potential interactions. Thus, the seemingly global dependences created by the timing derive from local causes.

In this paper, we propose a notion of time process for time Petri nets, examine the relation between the time processes and the firing schedules of the nets, and sketch algorithms for verifying validness of timings and constructing valid timings for a given process. The basic idea is that under time constraints, not all processes or timings of the events are possible. We find out which processes have a valid timing and which do not, and what kind of timings a process can have. The paper is a summary of [1].

2 Time Petri nets

The following section recalls the basic definitions of time Petri nets. The domain of time values $\mathbb{T}$ is the set of natural numbers. We denote by $[\tau_1, \tau_2]$ the closed interval between two time values $\tau_1, \tau_2 \in \mathbb{T}$, and by $\mathbb{I}$ the set of such intervals. Infinity is allowed at the upper bound. An interval can be of zero length ($\tau_1 = \tau_2$), containing only a single time value.

Definition 1. A *time Petri net* is a five-tuple $TPN = (P, T, F, SI, M_0)$, where P is a set of *places*, T is a set of *transitions*, $P \cap T = \emptyset$, $F \subseteq (P \times T) \cup (T \times P)$ is a *flow relation*, $SI : T \longrightarrow \mathbb{I}$ is a function called *static interval*, and $M_0 \subseteq P$ is the *initial marking* of the time Petri net. The tuple (P, T, F, M_0) is the *underlying net*. The boundaries of the static interval are called *earliest firing time* Eft and *latest firing time* Lft. The *preset* of $x \in P \cup T$ is ${}^\bullet x = \{y \mid yFx\}$ and *postset* is $x^\bullet = \{y \mid xFy\}$. It is assumed that the initial marking is finite, $|M_0| < \infty$, and that there is some constant *branching factor* $\beta < \infty$ such that $|{}^\bullet x| < \beta$ and $|x^\bullet| < \beta$ for all $x \in P \cup T$. Moreover, the presets and postsets of transitions must be nonempty, $|{}^\bullet t| > 0$ and $|t^\bullet| > 0$ for all $t \in T$.

Time Petri nets were introduced in [10]. The above definition defines time Petri nets as elementary net systems enriched with the static intervals on transitions, and with some finiteness requirements. As noted in [14], for finite nets, rational intervals can be converted to integers by multiplying the boundary values by the least common multiple of their denominators. The fundamental concepts of net behavior like enabledness and firing will defined next.

Definition 2. A transition t of a time Petri net is *enabled* at marking M iff ${}^\bullet t \subseteq M$. The set of all enabled transitions at marking M is denoted by $\mathit{Enabled}(M)$.

In time Petri nets, a marking is not sufficient information to describe a complete state of the system. The state must also include timing information. This is given as a clock function that, for each enabled transition, gives the amount of time that has passed since it has become enabled.

Definition 3. A *state* of a time Petri net $TPN = (P, T, F, SI, M_0)$ is a pair $S = (M, I)$, where M is a marking of TPN, and $I : \mathit{Enabled}(M) \longrightarrow \mathbb{T}$ is called the *clock function.* The *initial state* of TPN is $S_0 = (M_0, I_0)$, where $I_0(t) = 0$ for all $t \in \mathit{Enabled}(M_0)$.

For the firing of a transition to be possible at a certain time, four conditions must be satisfied.

Definition 4. A transition t may fire from state $S = (M, I)$ after delay $\theta \in \mathbb{T}$ iff

$$t \in \mathit{Enabled}(M), \tag{1}$$

$$(M \setminus {}^\bullet t) \cap t^\bullet = \emptyset, \tag{2}$$

$$Eft(t) \leq I(t) + \theta, \text{ and} \tag{3}$$

$$\forall t' \in \mathit{Enabled}(M) : I(t') + \theta \leq Lft(t'). \tag{4}$$

A transition satisfying Equations 1,3 and 4 but not Eqn. 2 is said to be *in contact.* The set of all transitions that may fire from state S is denoted by $\mathit{Fireable}(S)$ and the set of all transitions in contact is $\mathit{Contact}(S)$. From the definition we see directly that $\mathit{Fireable}((M, I)) \subseteq \mathit{Enabled}(M)$. Not all enabled transitions can fire because of timing constraints. The new marking after a firing is calculated as follows:

Definition 5. When transition t *fires after time* θ from state $S = (M, I)$, the new state $S' = (M'', I')$ is given as follows:

$$M' = M \setminus {}^\bullet t, \tag{5}$$

$$M'' = M' \cup t^\bullet, \text{ and} \tag{6}$$

$$I'(t) = \begin{cases} I(t) + \theta & \text{if } t \in \mathit{Enabled}(M'), \\ 0 & \text{if } t \in \mathit{Enabled}(M'') \setminus \mathit{Enabled}(M'), \\ \text{undefined} & \text{else.} \end{cases} \tag{7}$$

In time Petri nets, the firing sequence is enriched with timing information, and it is called firing schedule [3].

Definition 6. A *firing schedule* of a time Petri net is a finite or infinite sequence of pairs of transitions and time values

$$\sigma = (t_1, \theta_1), (t_2, \theta_2), (t_3, \theta_3), \ldots \qquad (8)$$

where the t_i are transitions and $\theta_i \in \mathbb{T}$ are their *firing delays.* The firing schedule is *fireable* from the initial state S_0 if there exist states $S_1, S_2, S_3, \ldots$ such that the transition t_i may fire from state S_{i-1} (according to firing condition in Def. 4) and the firing leads to new state S_i (according to the firing rule in Def. 5) for $i = 1, 2, 3, \ldots$. A state of a time Petri net is *reachable* if some fireable firing schedule leads from the initial state of the net to that state. A marking is reachable iff there is a reachable state with that marking.

We want to exclude from our class of nets the ones where Eqn. 2, is needed to prevent a second token from being inserted to a place.

Definition 7. The time Petri net is *contact-free* iff in every reachable state S of the net, no transition is in contact, $\mathit{Contact}(S) = \emptyset$.

Assumption 1. From now on, all time Petri nets will be assumed contact-free.

Another anomaly that we need to exclude from the class of nets under consideration is the possibility of infinite firing schedules that consume no time. Moreover, in infinite nets there can be infinite firing schedules where transitions consume nonzero amounts of time, but the the total time is bounded by a finite constant.

Definition 8. The time Petri net TPN has *divergent time* iff for every infinite firing schedule $(t_1, \theta_1), (t_2, \theta_2), (t_3, \theta_3), \ldots$ of TPN, the series $\theta_1 + \theta_2 + \theta_3 + \cdots$ diverges.

Assumption 2. From now on, we assume that all time Petri nets have divergent time.

3 Time processes

The firing schedule presentation of system behavior forces causally independent events in the system into a linear order. In this section we will define a presentation that retains both causal dependence and concurrency. Time processes of time

Petri nets will be constructed by labeling traditionally defined causal processes with time values and giving validness criteria for them.

We recall the definitions of a causal net and homomorphism from the literature. The definition of homomorphism from time Petri nets to causal nets is a straightforward adaptation of the usual net homomorphism. A causal process of a time Petri net is then defined as a causal net together with a homomorphism.

Definition 9. A *causal net* $CN = (B, E, G)$ is a finitary, acyclic net, where $\forall b \in B : |b^\bullet| \leq 1 \wedge |{}^\bullet b| \leq 1$. Places B of a causal net are called *conditions* and transitions E are called *events*. Preplaces are called *preconditions* and postplaces *postconditions*.

Finitary means that every $x \in B \cup E$ has only a finite number of G-predecessors. We denote by $\leq$ the partial order G^* on $B \cup E$ and by $<$ the corresponding strict partial order. Event e is said to causally precede e' if $e \leq e'$. We write $Min(CN)$ for the $\leq$-minimal elements of a causal net and $Max(CN)$ for the $\leq$-maximal events, and call these *initial elements* and *final elements*, respectively. In a nonempty net where G is finitary, there is always at least one minimal element. In a contact-free causal net, the minimal elements are places. In a causal net, we identify the preset and postset of a condition with the unique events in them and use the notations ${}^\bullet b$ and $b^\bullet$ to mean a single event.

The relation between time Petri nets and causal nets is be stated using a homomorphism, a mapping from the causal net to the underlying net of the time Petri net that preserves the local structure of the net.

Definition 10. Let $TPN = (P, T, F, SI, M_0)$ be a time Petri net and $CN = (B, E, G)$ a causal net. A mapping $p : B \cup E \longrightarrow P \cup T$ is a *homomorphism* if $p(B) \subseteq P$, $p(E) \subseteq T$, $\forall e \in E$: the restriction of p to ${}^\bullet e$ is a bijection between ${}^\bullet e$ and ${}^\bullet p(e)$ and the restriction to $e^\bullet$ is a bijection between $e^\bullet$ and $p(e)^\bullet$, and the restriction of p to $Min(CN)$ is a bijection between $Min(CN)$ and M_0.

For a downward closed set of events E', we define a function Cut by $Cut(E') = (E'^\bullet \cup Min(NS)) \setminus {}^\bullet E'$. The intuition of a cut is that it represents a state of the system.

Definition 11. Let $TPN = (P, T, F, SI, M_0)$ be a time Petri net. A *causal process* of TPN is a pair (CN, p), where CN is a causal net and p is a homomorphism from CN to TPN.

If $\forall b, b' \in B' : p(b) = p(b') \Rightarrow b = b'$, we say that B' *maps injectively* to places. In processes of 1-safe untimed net systems, sets $Cut(E')$ are guaranteed to map injectively to places for all downward closed sets E'.

The notion of a time process for a Time Petri net can now be defined. This is done by adding timing information to the causal processes of the time Petri net: time values depicting occurrence times are attached to the events of the process. Since time constraints imposed by the earliest and latest firing times of events restrict the set of possible timings the events of the time process can have, validness criteria are defined for deciding if the process with the time values is a *time process* of the time Petri net. As a result not all timings are possible and some causal processes may even have no valid timings at all, being thus impossible in the system.

In order to motivate the definition of valid timing, we demonstrate the dependences between events of a time Petri net. Figure 1 shows a segment of a time process of a time Petri net, that is a segment of a causal process of the time net, where the occurrence times are written next to the events. The dashed transitions are not part of the process, instead they denote transitions whose occurrence is not known yet, ie. the corresponding events have not been added to the process yet. Assume that the global clock has advanced to $time = 8$, and that we want to consider, whether we can add the event e with timing 8 to the process. The darker grey area is the causal past of e, and the light grey area contains the events with an earlier occurrence time than e. The upper black line intuitively marks the two possible global states at which event e can occur depending on the interleaving of concurrent events. The lower black line marks the global state before the clock was advanced the last time. Let us denote the state of the system on the lower line by C_e. Below, we will argue that the possibility of event e and its occurrence time should be determined based on the state C_e rather than the causal past of e or the global state at which e occurs.

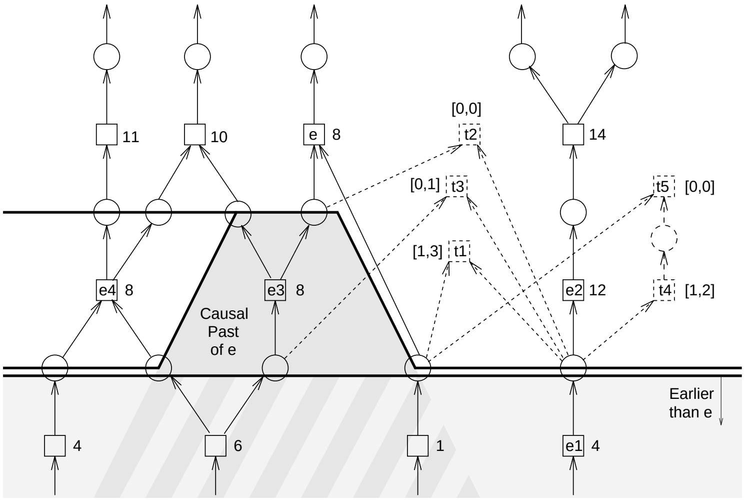

Fig. 1. Events and occurrence times

We want to show how dependences arise between causally unrelated parts of the process. Look at transition t_1 which is enabled at the global state where also e is enabled. It became enabled at time 4 and has to fire before or at time $4+3=7$. Therefore, event e cannot occur at time 8. However, if event e_2 would have a timing of 5 it could disable t_1 thus again making it possible for e to occur. In this way, events e_1 and e_2 and their timing can affect the occurrence of e. Thus, it is not enough to look at the local state where e is enabled, but one must also consider causally unrelated parts of the process in order to know if e can occur at the given time. This would indicate that we have to consider the global state where e is enabled. However, the state is not uniquely known, as demonstrated by the branching upper black line in the figure. Even though t_2 is enabled at the global state at which e is enabled, it cannot prevent e from occurring. This is because one of the conditions enabling t_2 is not in C_e and, consequently, the latest firing time of t_2 cannot be earlier than the occurrence time of e. Thus, when assessing the possibility of event e, it is not necessary to consider all conditions in the global state where e occurs, but only those in C_e. Transition t_3 is enabled at C_e but not at any of the possible global states where e occurs. It can still stop e from occuring, because it does not allow e_3 to occur and e depends causally on e_3. Now consider transition t_4: it has to fire at time 6 or earlier. But unlike the other transitions, t_4 does not have causal relation to e through its preset. Therefore, one might think that t_4 cannot in any way affect e. But this is not true! Indeed t_4 does stop e from occurring because its successor, the transition t_5 inevitably disables e. We might not know that there is a transition t_5, but to be on the safe side, we must require all transitions enabled at C_e to have latest possible firing times greater than or equal to the occurrence time of e. Otherwise, we cannot know that the occurrence time of e is possible. Events e, e_3 and e_4 have the same occurrence time and $C_e = C_{e_3} = C_{e_4}$. If we check that one of the events is not kept from firing by transitions enabled at C_e, it implies the same for all of them. Thus, all events with the same occurrence time can be processed together. The preceding discussion is now summarized in the following definition.

Definition 12. Let TPN be a time Petri net and (CN, p) its causal process, where $CN = (B, E, G)$. A *timing function* $\tau : E \longrightarrow \mathbb{T}$ is a function from events into time values. The values of τ are called *occurrence times* of the events. If B' is a set of conditions and transition t is enabled at $p(B')$, the *time of enabling* for t in B' is defined as

$$TOE(B', t) = \max(\{\tau({}^\bullet b) \mid b \in B' \setminus Min(CN) \wedge p(b) \in {}^\bullet t\} \cup \{0\}). \tag{9}$$

The set of *earlier events* for an event e is

$$Earlier(e) = \{e' \in E \mid \tau(e') < \tau(e)\}\,. \tag{10}$$

A timing function τ is a *valid timing* of the causal process iff

$$\forall e \in E : \tau(e) \geq TOE({}^\bullet e, p(e)) + Eft(p(e)), \text{ and} \tag{11}$$

$$\forall e \in E :\ \forall t \in Enabled(p(C_e)) : \tau(e) \leq TOE(C_e, t) + Lft(t), \tag{12}$$

where $C_e = Cut(Earlier(e))$.

A *time process* of TPN is a triple (CN, p, τ) where (CN, p) is a causal process of TPN and τ is a valid timing of the causal process.

The definition of valid timing has been derived from the firing condition of time Petri nets (Def. 4). The two criteria (Eqn. 11 and 12) impose the earliest and the latest firing times on the events (like Eqn. 3 and 4, respectively). The auxiliary function TOE gives the time when a transition becomes enabled at a set of conditions, i.e. the occurrence time of the last of the previous events. When there are no previous events (t is enabled at the initial marking), the time of enabling is naturally zero. The lower bounds of occurrence times (Eqn. 11) are easily checked as they depend only on the previous events. The upper bounds (Eqn. 12) are more complicated because they create dependences between causally unrelated parts of the process, as seen in the example above. Thus the latest firing times of all transitions enabled at the set $C_e = Cut(Earlier(e))$ must be considered. This is the set depicted by the lower black line in figure 1. By utilizing the above definitions, it is possible to derive an algorithm that checks the validness of a timing function in time $O(N \log N + \beta^2 N|P|)$, where N is the number of different values of the timing function in the process, β is the branching factor and $|P|$ the number of places in the net (cf. Sec. 3.3 in [1]).

In Figure 2, there is a time Petri net and its time process. ($\mathtt{w} = \infty$.) The values of the timing function have been printed next to the events. The timing in Fig. 2(b) is valid. All different sets $Cut(Earlier(e))$ are shown in grey. There are only three such sets because two are equal: $\tau(e_2) = \tau(e_3)$ implies $Cut(Earlier(e_2)) = Cut(Earlier(e_3))$. The transitions enabled at the sets have been drawn with dashed lines. On the other hand, the timings in Figure 3 are not valid. The sets and enabled transitions that conflict with Eqn. 12 are shown. In (a), transition t_2 should fire before e_3 and disable it. The timing is clearly not in accordance with the firing condition. Net (b) demonstrates a more surprising requirement for validness. In Fig. 2(b) we already saw that the firing times of the transitions are completely legal. Nevertheless, the timing is not valid. This is because the left side of the process has not been generated far enough to make conclusions about the possibility of the firing times. We will see in Section 5 that it is essential for all parts of the process to be complete up to same time value. Otherwise, validness cannot be determined.

4 Processes and firing schedules

In this section, we will look more closely at the relation between time processes and firing schedules. A time process is a partial order that can be sorted into several different linear orders. The linear orders that respect the occurrence times of the events will be called interleavings of the time process. We will show that

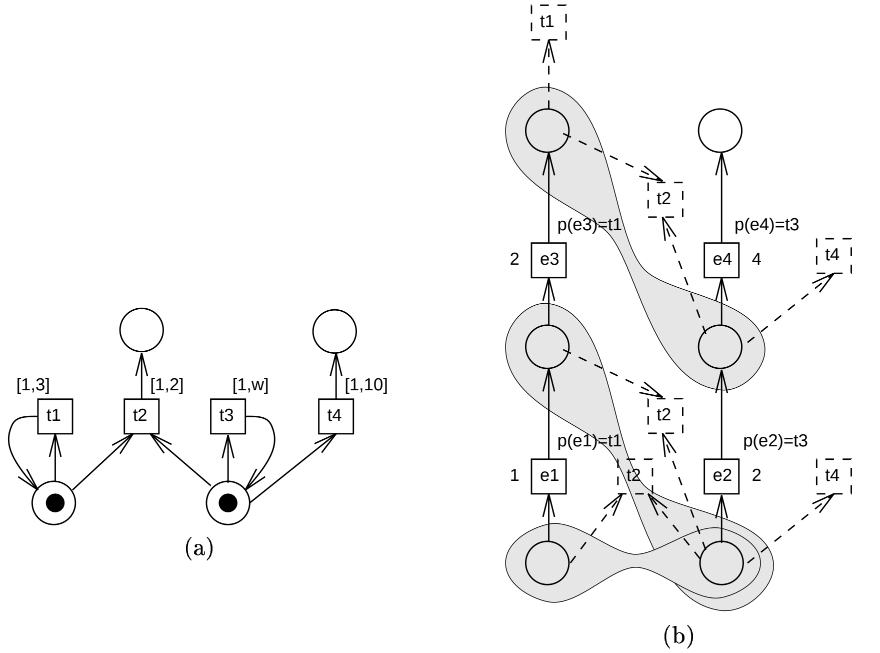

Fig. 2. A time Petri net and a process with valid timing

in a time Petri net, there is one-to-one correspondence between interleavings of time processes and firing schedules that are fireable from the initial state of the net.

We want to define interleaving of a time process as a linearization of the partial order of events where the events are ordered also by their occurrence times. In order for this to make sense, we have to prove that the causal and time order never conflict with each other.

Lemma 13. *Let (CN, p, τ) be a time process of a time Petri net, where $CN = (B, E, G)$. Let also $e, e' \in E$ be two events of the process. Then,*

$$e \leq e' \;\Rightarrow\; \tau(e) \leq \tau(e') \tag{13}$$

It is now possible to give the definition of an interleaving of a time process. A function from interleavings to firing schedules will also be defined. Interleavings and the function will link time processes to firing schedules.

Definition 14. An *interleaving* of a time process is a finite or infinite sequence $\rho = e_1, e_2, e_3, \ldots$ consisting of the events of the process, such that every event is

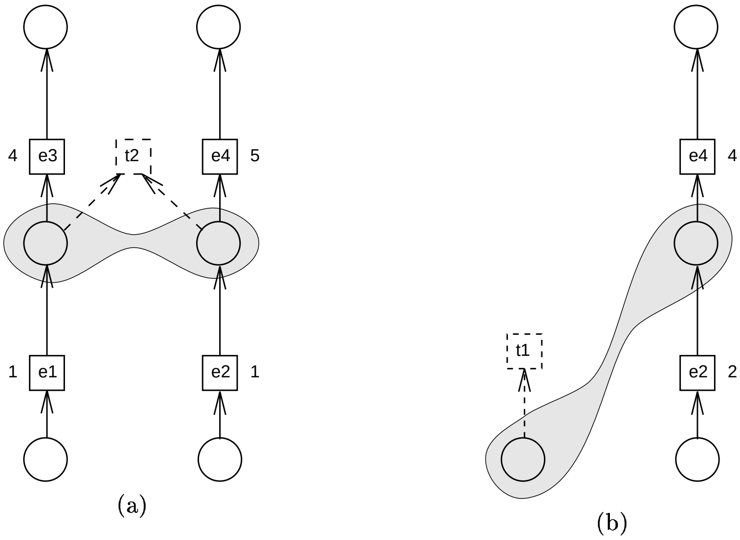

Fig. 3. Two invalid timings

in the sequence exactly once, and both causal and time order are preserved,

$$(e_i < e_j \ \vee \ \tau(e_i) < \tau(e_j)) \ \Rightarrow \ i < j \text{ for all } i, j. \tag{14}$$

The function FS maps interleavings to firing schedules of the net.

$$FS(\rho) = (p(e_1), \tau(e_1) - 0), \ (p(e_2), \tau(e_2) - \tau(e_1)), \ \ldots \tag{15}$$

To summarize, we have now three different order relations on the set of events: causal order ($e < e'$), time order ($\tau(e) < \tau(e')$) and a linear order in an interleaving ($e_i < e_{i+1}$). The orders do not conflict with each other. Causal and time order both imply order in interleavings.

In the rest of this section, we will give theorems that state that the function FS is a bijective mapping from the interleavings of the processes of a time Petri net to the firing schedules fireable from the initial state of the net. For details of the proofs we refer to [1].

The first important property of FS is that interleavings of a time process are mapped to firing schedules that are fireable from the initial state of the net.

Theorem 15. *If ρ is an interleaving of a time process of a time Petri net, then the firing schedule $FS(\rho)$ is fireable from the initial state of the net. The markings in the intermediate states of the firing schedule are $M_k = p(Cut(E_k))$ for $k = 1, 2, 3, \ldots$.*

Proof. See Thm. 21 [1], p. 29.

We know now that there is a fireable firing schedule corresponding to every interleaving.

Theorem 16. *Let TPN be a time Petri net with divergent time. Every time process of TPN has an interleaving.*

Proof. See Thm. 22 [1], p. 32.

The firing schedules in the range of the function FS are known to be fireable from the initial state of the time Petri net in question. We still have to show that the function is a bijection. The next theorem will show that it is surjective.

Theorem 17. *Given a firing schedule σ of a time Petri net TPN, it is possible to construct a process of TPN with interleaving ρ, such that $\sigma = FS(\rho)$.*

Proof. See Thm. 23 [1], p. 34.

So far, we have shown that there is a surjective mapping FS from the interleavings of time processes to the firing schedules fireable from the initial state of the net. The next theorem will complete the proof that the FS is bijective.

Theorem 18. *Let σ be a firing schedule of a time Petri net TPN. The process of TPN with interleaving ρ, such that $\sigma = FS(\rho)$, is unique up to renaming of elements.*

Proof. See Thm. 24 [1], p. 37.

The results in this section tell us, that in a contact-free time Petri net, the relation between firing schedules fireable from the initial state and interleavings of time processes is bijective. These results are not surprising, but they assure that the time processes correctly represent the behavior of the system.

5 All valid timings of a process

It would be desirable to somehow characterize the set of all valid timings for a given process. The definition of valid timing only gives a way to check the validness, but it does not help much in constructing timing functions. Still, the need of having the entire set of valid timings is obvious. From the set, one can answer questions like, what is the longest time that can pass between two events, can an event occur after another, and so on. We will see that the set of all timings can be computed in a fairly general setting, but that the cost of the computation may be high.

We will start by giving an alternative characterization of valid timings. It is based on the idea that in a process a decision has to be made about the firing or the disabling of all potentially enabled transitions and that this decision is observed in the process as the occurrence of an event in the system. As the approach of this section is aimed at computing the set of all of valid timings, it cannot take advantage of the structure of a single timing function like the definition of validness does.

In processes of untimed net systems, all antichains of conditions represent subsets of reachable markings. This is not the case in processes of time Petri nets, but the antichains are still useful, because all reachable markings are represented by some of them. Antichains are traditionally called co-sets, *co* meaning a concurrency relation between process elements [4].

Definition 19. A set B' of conditions in a causal net is a *co-set* if no two conditions in the set are causally related.

$$\forall b, b' \in B' : (\neg(p \leq q) \wedge \neg(q \leq p)) \tag{16}$$

A maximal co-set is called a *cut.*

Recall that a transition in a time Petri net may be enabled but not fireable (cf. Def. 4). When a transition becomes enabled in a time Petri net, it will eventually fire by its latest firing time, unless it is disabled by the firing of another transition. However, it is also possible for the disabling to occur before the transition becomes fireable. Since our definition does not require a processes to continue infinitely or until all enabled transitions have been fired, a process is not necessarily maximal. The first transitions left out at the end of the process, ie. whose corresponding events have not yet been added to the process, are naturally enabled at the marking corresponding to $Cut(E)$. In the next definition, these transitions are called *extension transitions.* Whether an extension transition t is fired or not depends on whether it is in conflict with some other transition t', in which case there is a *choice* between t or t'. This choice is resolved or *decided* by an event e, because when we add the event e to the process, we know which one of the transitions was fired.

Definition 20. Let (CN, p) be a causal process of a time Petri net, where $CN = (B, E, G)$. Transition t is a *choice transition* at $B' \subseteq B$ iff B' is a co-set that maps injectively to places and ${}^\bullet t = p(B')$. Transition t is an *extension transition* iff $B' \subseteq Cut(E)$.

Let e be an event and t a choice transition at B' with $B' \cap {}^\bullet e \neq \emptyset$ and

$$\tau(e) \leq TOE(B', t) + Lft(t). \tag{17}$$

Then, e *decides* t at B, or e is a *deciding event* of t.

The deciding event of choice transition t at B' is either a firing of t, or it is the firing of another transition that disables t. In Figure 4, there is a time Petri net and its process. At the initial marking, $t1$ and $t2$ are choice transitions. Both are decided by the event $e1$; $t1$ is fired and $t2$ is disabled. $t3$ is an extension transition and remains undecided. In this case, $t2$ never becomes fireable, since it is disabled before its earliest firing time.

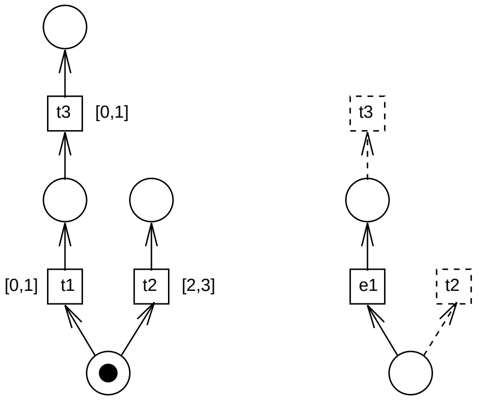

Fig. 4. Choice and extension transitions

Definition 21. A causal process (CN, p) of a time Petri net, where $CN = (B, E, G)$, is *complete* with respect to timing function τ iff for every extension transition t of the process,

$$\max\{\tau(e) \mid e \in E\} \;\leq\; TOE(Cut(E), t) + Lft(t). \tag{18}$$

When the process is complete, none of its concurrent parts "are left behind in time". That is, if there is some event in the process with time $\tau(e)$, all potential events with smaller value of τ must be included in the process. This property is also implicit in the definition of valid timing.

It is computationally more efficient to state completeness in terms of the final conditions of the process.

Lemma 22. *If Eqn. 11 holds, Eqn. 18 is equivalent with*

$$\forall e \in E \,:\, ((e^{\bullet\bullet} = \emptyset \,\wedge\, e \not\in {}^\bullet B') \;\Rightarrow\; \tau(e) \leq TOE(Cut(E), t) + Lft(t)). \tag{19}$$

where $B' \subseteq Cut(E)$ is the set of conditions at which t is enabled.

Proof. When Eqn. 11 holds, the causal order of events implies time order, $e' \leq e \Rightarrow \tau(e') \leq \tau(e)$. Then, the maximum value of τ is obtained for some of the the causally maximal events in the causal order. Hence, it is enough to examine only those events in the causal order, for which $e^{\bullet\bullet} = \emptyset$. Moreover, if $e \in B' \subseteq {}^{\bullet}Cut(E)$, then $\tau(e) \leq TOE(Cut(E), t)$, and the inequality holds automatically.

Before discussing how to construct the set of all valid timings, we state the main theorem of this section. It characterizes the validness of a timing in an alternative way. While the definition of validness is optimized for checking known timing functions, this formulation of the validness criteria makes it possible to talk about sets of timings.

Theorem 23. *Let TPN be a time Petri net and (CN,p) a causal process of TPN. A timing function τ is valid iff the validness criterion on the earliest firing time (Eqn. 11) holds, the process is complete with respect to the timing, and every choice transition is either an extension or decided by some event in the process.*

Proof. See Thm. 30 [1], p.41.

A closer look at Theorem 23 reveals that all the properties required by the theorem from valid timings can be presented with inequalities. The three sources for inequalities are:

1. validness criterion on the earliest firing time (Eqn. 11),
2. deciding events of choice transitions (Eqn. 17),
3. completeness of the process (Eqn. 19).

Source 1 gives a set of inequalities on occurrence times that must be satisfied by all valid timings. Source 3 produces inequalities with the function TOE in them. Since TOE is defined as a maximum over a set of events, these can be expanded to sets of alternative inequalities with only occurrence times as variables. At least one of the alternatives in each set must hold in every valid timing. The same applies to Source 3, but these inequalities have to be instantiated with all the events satisfying the left side of the implication in Eqn. 19.

The existence of a valid timing for a causal process can be determined by an algorithm that nondeterministically chooses elements from each set of alternative inequalities. If any one of the nondeterministic choices results in a set of inequalities with a feasible solution, the solution is a valid timing. The resulting inequalities are all linear and there are up to $(|T||B|^{\beta} + \beta)|E|$ of them. In addition, all variables $\tau(e)$ are nonnegative. The existence of a feasible solution for the set of a polynomial number of linear inequalities is decidable in polynomial time [5]. There exists a valid timing for the process iff the set of inequalities IE in one of the nondeterministic execution paths of the algorithm has a solution. If the branching factor β is constant for a class of Petri nets, then it is possible to decide in nondeterministic polynomial time the existence of a valid timing. We state this as a theorem.

Theorem 24. *In a class of time Petri nets where the branching factor β is bounded by a constant, the existence of a valid timing can be decided by an algorithm that is NP with respect to the size of the process.*

Let us now turn to the question of when an invalid timing can't be extended to a valid one. (CN', p') is an *initial subprocess* of (CN, p) iff $B' \cup E'$ is a downward closed (with respect to G) subset of $B \cup E$ and G' and p' are restrictions of G and p to $B' \cup E'$.

Theorem 25. *Let (CN, p), where $CN = (B, E, G)$, be a causal process with timing function τ. Assume that there is a choice transition t at a set B_t that is neither an extension nor decided by any event of E. If (CN, p) is an initial subprocess of another process $(\hat{CN}, \hat{p})$ such that ${B_t}^{\bullet}$ is equal in both processes, then $(\hat{CN}, \hat{p})$ does not have any valid timing function whose restriction to E equals τ.*

Proof. It is easy to see that the candidates for deciding events are the same in both processes. If none is added, the process remains invalid.

The most common case where ${B_t}^{\bullet}$ is equal in both processes is the one where $b^{\bullet} \in E$ for all $b \in B_t$. This is because when all conditions of the set B_t are consumed by some events in the smaller process, there is not room to add any new deciding events in the larger one. Consequently the invalid situation cannot be corrected by extending the process. This is the case for the process of Fig. 3(a).

In Figure 5, there is a causal process of the time Petri net of Fig. 2(a), with all choice transitions. We are interested in the set of valid timings for this process. The set of inequalities is given in figure 6. The first four constraints are the earliest firing times of the events. The next two guarantee the completeness of the process. Then come the alternative reasons why each choice transition does not reach its latest firing time. Four of the choices are decided by events that are firings of the choice transition.

These inequalities have a solution and, thus, there exists a valid timing. We can, for instance, find the maximum of $\tau(e_3) - \tau(e_2)$. The maximum is 5, at

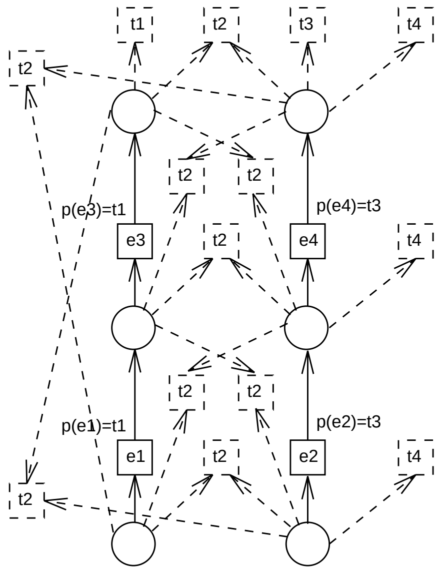

Fig. 5. Computing all timings of a process

$\tau(e_1) = 3$, $\tau(e_2) = 1$, $\tau(e_3) = 6, \tau(e_4) = 4$. We can also find maximum firing time of event e_4. The result is 7, at $\tau(e_1) = 2$, $\tau(e_2) = 6$, $\tau(e_3) = 4, \tau(e_4) = 7$. The latter result may appear surprising, but in order for $\tau(e_4)$ to be higher, other parts of the process should be extended. Thus, these results are only a maximum in this process, not necessarily in larger processes.

The set of constraints can be reduced by keeping only those constraints that are true in all larger processes. The reduced set of constraints is given in figure 7. In this set, the maximum of $\tau(e_3) - \tau(e_2)$ is still 5. The result indicates that it is the maximum in all processes having the process of Fig. 5 as an initial subprocess. On the other hand, the maximum of e_4 differs considerably from the previous result. It is now 20, obtained for example at $\tau(e_1) = 1, \tau(e_2) = 10, \tau(e_3) = 2, \tau(e_4) = 20$. In order to learn the maximum firing time of event e_4, we would have consider all different ways in which the process can be extended. The value obtained from the reduced set of inequalities, 20, is only an upper bound for $\tau(e_4)$.

6 Conclusions

The objective of this work was to give time Petri nets a partial order semantics. As the main contribution, we introduced time processes for representing the causal behavior of contact-free time Petri nets. Time processes were defined as causal processes with a valid timing on the events. The relationship between time processes and the firing schedule semantics was examined with the conclusion

$$\begin{array}{ll}
\quad(\tau(e_1) \geq 1) & \text{Eft} \\
\wedge\ (\tau(e_3) - \tau(e_1) \geq 1) & \\
\wedge\ (\tau(e_2) \geq 1) & \\
\wedge\ (\tau(e_4) - \tau(e_2) \geq 1) & \\
\wedge\ (\tau(e_4) - \tau(e_3) \leq 3) & \text{Lft: extensions} \\
\wedge\ (\tau(e_3) - \tau(e_4) \leq 10) & \\
\wedge\ (\tau(e_1) \leq 3) & \text{Lft: events} \\
\wedge\ (\tau(e_3) - \tau(e_1) \leq 3) & \\
\wedge\ (\tau(e_2) \leq 10) & \\
\wedge\ (\tau(e_4) - \tau(e_2) \leq 10) & \\
\wedge\ (\tau(e_1) \leq 2\ \vee\ \tau(e_2) \leq 2) & \text{Lft: other choices} \\
\multicolumn{2}{l}{\wedge\ (\tau(e_3) - \tau(e_1) \leq 2\ \vee\ \tau(e_3) - \tau(e_2) \leq 2\ \vee\ \tau(e_4) - \tau(e_1) \leq 2\ \vee\ \tau(e_4) - \tau(e_2) \leq 2)} \\
\wedge\ (\tau(e_1) - \tau(e_2) \leq 2\ \vee\ \tau(e_4) - \tau(e_2) \leq 2) & \\
\wedge\ (\tau(e_3) - \tau(e_1) \leq 2\ \vee\ \tau(e_2) - \tau(e_1) \leq 2) & \\
\wedge\ (\tau(e_3) - \tau(e_1) \leq 2\ \vee\ \tau(e_3) - \tau(e_4) \leq 2) & \\
\wedge\ (\tau(e_4) - \tau(e_2) \leq 2\ \vee\ \tau(e_4) - \tau(e_3) \leq 2) & \\
\wedge\ (\tau(e_2) - \tau(e_3) \leq 2) & \\
\wedge\ (\tau(e_1) - \tau(e_4) \leq 10) &
\end{array}$$

Fig. 6. The set of inequalities generated by the process in 5

that the approaches are compatible. We then sketched algorithms for checking validness of known timings and for constructing the set of all valid timings of a process. The complexity of timing analysis lies in dependences between causally unrelated events, created by the latest firing times of transitions. With the partial order approach, these dependences must be explicitly dealt with. They forced us to look at other enabled transitions, not only the causal past of an event, when assessing validness of the occurrence time of the event.

The definition of valid timing was optimized for checking validness of known timings. As such, it did not give any direct way for constructing time processes. Therefore, an alternative characterization for validness was presented. With the alternative formulation of the validness criteria, the existence of a valid timing for a given process can be decided in nondeterministic polynomial time. The algorithm can also be used for constructing the set of all valid timings. This set is presented as sets of alternative linear inequalities, and with these, one can answer questions like, what is the maximal time separation between two events.

The presented techniques could be improved with various optimizations, for example, in solving the sets of linear inequalities. In a net class with restricted forms of synchronization, eg. extended free choice time Petri nets, the validness criteria could be simplified significantly. Also, other restricted net classes could

$$
\begin{array}{ll}
\ (\tau(e_1) \geq 1) & \text{Eft} \\
\wedge\ (\tau(e_3) - \tau(e_1) \geq 1) & \\
\wedge\ (\tau(e_2) \geq 1) & \\
\wedge\ (\tau(e_4) - \tau(e_2) \geq 1) & \\
\wedge\ (\tau(e_1) \leq 3) & \text{Lft: events} \\
\wedge\ (\tau(e_3) - \tau(e_1) \leq 3) & \\
\wedge\ (\tau(e_2) \leq 10) & \\
\wedge\ (\tau(e_4) - \tau(e_2) \leq 10) & \\
\wedge\ (\tau(e_1) \leq 2\ \vee\ \tau(e_2) \leq 2) & \text{Lft: other choices} \\
\multicolumn{2}{l}{\wedge\ (\tau(e_3) - \tau(e_1) \leq 2\ \vee\ \tau(e_3) - \tau(e_2) \leq 2\ \vee\ \tau(e_4) - \tau(e_1) \leq 2\ \vee\ \tau(e_4) - \tau(e_2) \leq 2)} \\
\wedge\ (\tau(e_1) - \tau(e_2) \leq 2\ \vee\ \tau(e_4) - \tau(e_2) \leq 2) & \\
\wedge\ (\tau(e_3) - \tau(e_1) \leq 2\ \vee\ \tau(e_2) - \tau(e_1) \leq 2) &
\end{array}
$$

Fig. 7. The reduced set of inequalities

be considered. Interesting topics for future research include incorporating the timing analysis with process generation to compute properties of whole nets, not only single processes, and extending the presented methods to time stream Petri nets. Also, the idea of branching processes for time Petri nets is intriguing, although it is difficult to see how the different time processes could be embedded into one partial order.

References

1. Tuomas Aura. Time processes of time petri nets. Technical Report A38, Digital Systems Laboratory, Helsinki University of Technology, Espoo, Finland, 1996. Available through `http://saturn.hut.fi/pub/reports/A38abstract.html`.
2. Bernard Berthomieu and Michel Diaz. Modelling and verification of time dependent systems using time Petri nets. *IEEE Transactions on Software Engineering*, 17(3):259–273, 1991.
3. Bernard Berthomieu and Miguel Menasche. A state enumeration approach for analyzing time Petri nets. In *Proceedings of the 3rd European Workshop on Applications and Theory of Petri Nets*, pages 27–56, September 1982.
4. Eike Best and César Fernández. *Nonsequential Processes, A Petri Net View*, volume 13 of *EATCS Monographs on Computer Science*. Springer-Verlag, 1988.
5. Thomas H. Cormen, Charles E Leiserson, and Ronald L. Rivest. *Introduction to algorithms*. MIT Press, 1990.
6. Michel Diaz and Patrick Sénac. Time stream Petri nets, a model for timed multimedia information. In *Proceedings of the 15th International Conference on Application and Theory of Petri Nets 1994*, volume 815 of *LNCS*, pages 219–238. Springer-Verlag, June 1994.

7. Joost Engelfriet. Branching processes of Petri nets. *Acta Informatica*, 28:575–591, 1991.
8. Javier Esparza. Model checking using net unfoldings. *Science of Computer Programming*, 23:151–195, 1994.
9. Henrik Hulgaard and Steven M. Burns. Efficient timing analysis of a class of Petri nets. In *Computer Aided Verification 7th International Workshop CAV'95*, volume 939 of *LNCS*, pages 423–436. Springer-Verlag, 1995.
10. Philip M. Merlin and David J. Farber. Recoverability of communication protocols — implications of a theoretical study. *IEEE Transactions on Communications*, 24(9):1036–1043, 1976.
11. Tadao Murata. Petri nets, properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–580, 1989.
12. ISO/IEC JTC1/SC21/WG1 N1053. Enhancements to LOTOS, 1995.
13. J. L. Peterson. *Petri Net Theory and the Modeling of Systems.* Prentice-Hall, Inc., Englewood, NJ, 1981.
14. Louchka Popova. On time Petri nets. *Journal Inform. Process. Cybern. EIK*, 27(4):227–244, 1991.
15. C. Ramchandani. Analysis of asynchronous concurrent systems by timed Petri nets. Technical report, Project MAC, TR 120, MIT, February 1974.
16. Wolfgang Reisig. *Petri Nets: An Introduction*, volume 4 of *EATCS Monographs on Theoretical Computer Science.* Springer-Verlag, 1985.
17. Valentin Valero, David de Frutos, and Fernando Cuartero. Timed processes of timed Petri nets. In *Proceedings of the 16th International Conference on Application and Theory of Petri Nets 1995*, volume 935 of *LNCS.* Springer-Verlag, June 1995.
18. Józef Winkowski. Algebras of processes of timed Petri nets. In *CONCUR'94: Concurrency Theory*, volume 836 of *LNCS*, pages 195–209. Springer-Verlag, August 1994.