Distributed Semi-Markov Processes in Stochastic T-Timed Petri Nets

Stefan Haar

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Distributed Semi-Markov Processes in Stochastic T-Timed Petri Nets

Stefan Haar

N˚4754

March 2003

THÈME 4

ISSN 0249-6399 ISRN INRIA/RR--4754--FR+ENG

INRIA
RENNES

Distributed Semi-Markov Processes in Stochastic T-Timed Petri Nets

Stefan Haar*

Thème 4 — Simulation et optimisation
de systèmes complexes
Projets SIGMA2

Rapport de recherche n ° 4754 — March 2003 — 16 pages

Abstract: The *heaps of pieces* modelling approach (see Gaubert/Mairesse, PNPM'99) admits a (max,+)-linear model for the time consumption, under earliest firing and a given trace, of safe T-timed nets. The present paper shows that this type of model can be extended, using an appropriate partial order semantics under cluster view, to include stochastic choice and timing; we give the algorithmic construction of that semantics and obtain a semi-Markov property in multi-dimensional real time.

Key-words: stochastic T-timed nets, conflict clusters, partial order semantics, max-plus-algebra, semi-Markov processes

(Résumé : tsvp)

* Supported by the MAGDA2 project, RNRT; see the sites `http://www.telecom.gouv.fr/rnrt/index_net.htm` and `http://www.magda.elibel.tm.fr` for more information.

Unité de recherche INRIA Rennes

Dépliages probabilistes à Clusters des Réseaux de Pétri

Résumé : L'approche de modélisations par tas de pièces, voir Gaubert/Mairesse, PNPM'99, admet un modèle (max,+)-lineaire pour la consommation de temps, sous la politique du tir au plus tôt et trace d'événement donnée, pour les réseaux T-temporisés saufs. Le présent article montre que ce type de modèle peut être étendu, en utilisant une sémantique d'ordre partiel adaptée qui prend une perspective de clusters, pour inclure un choix et une temporisation stochastiques. Nous donnons la construction algorithmique de cette sémantique, et obtenons une proprieté sémi-Markovienne en temps réel multidimensionel.

Mots-clé : Réseaux de Petri T-temporisés, clusters, sémantique d'ordre partiel, algèbre max-plus, processus sémi-Markoviens

1 Introduction

In T-timed nets, transitions consume time when firing; the corresponding tokens are consumed at the *beginning* of the event. This contrasts with the *race policy* in *Stochastic Petri Nets (SPN)* where the transitions fire after a delay which is exponentially distributed (in *Generalized SPN* or *GSPN* [1, 2, 19], some transitions may be exempted from that rule). The conflict between several transitions competing for the same token starts when the enabling tokens arrive and is decided at the moment when the first competitor reaches the end of its random delay; only at that point is the token consumed. With probability one, no two delays end at the same moment, so the conflict is resolved indeed by this policy in a purely exponential setting; for generalizations, in which some transitions are untimed and may enter in a conflict not decidable by the race policy, routing or *weight*-defined decision probability measures have been proposed (see, e.g., [1, 2]). The behaviour of SPNs is described by a continuous time Markov chain (CTMC); the memoryless property of the exponential law is crucial for this. In more general settings, where firing times may have memory, one will typically not find CTMC's. Rather, one tries to establish a *regenerative*, *renewal*, or *Semi-Markov property*; it states, informally, the chain of markings is certain to reach, in finite time, a fixed marking state, at which point in time the process probabilistically restarts. As a future application (not carried out here), one can then use arguments from renewal theory to obtain existence of, e.g., asymptotic throughputs.

In [14], this was studied for life and bounded Free-Choice Petri nets *(l.b. FCN)* with probabilistic *routing* and arbitrary firing time distributions. *Routing* means that tokens produced on some place p are assigned, with associated probabilities, to one of the post-transitions $t \in p^\bullet$ (no routing is done when $|p^\bullet| = 1$); the free choice property then ensures that t will be enabled by that token (since t depends on no other place). It was shows that l.b. FCN have the following property: Suppose one transition b is chosen arbitrarily, but such that b is not in structural conflict with any other transition (i.e. the places in ${}^\bullet b$ have b as only outgoing transition). When b is *blocked* , that is, prevented from firing, then net will almost surely enter (possibly after firing many other transitions) a marking M_b in which no transition is enabled but b, and halt; and there is only one M_b with this property, i.e. the blocked marking is unique given b. Now, if the firing time distributions have "heavy enough tails", i.e. are such that with positive probability, b takes so much firing time that all other enabled transitions have finished their firings before b is done, the net is necessarily in the marking M_b' obtained from M_b by subtracting the tokens being consumed by b. Then, at the moment b finishes its firing, the process restarts probabilistically from the new marking and all clocks at 0.

The present work aims at extending the regeneration property to general Petri net topologies. Note first of all that no direct extension of the approach in [14] is possible, since the key results fail to hold outside the class of FCN (or rather a certain proper superset of FCN, containing those nets that "behave like FCN"; see the discussion in [14]). In fact, the approach we propose here combines partial order semantics, generally used to handle concurrency in untimed nets, with *physical* time. In fact, we will treat physical time not as a scalar, but multi-dimensional object, reflecting the distributed and local rather than global nature of PN evolution. We there extend the *heap model* approach of Gaubert and Mairesse [12, 13]; see also [23]. There, partial orders and physical time are brought together successfully, using the analogy observed by Viennot [24] of heap monoids with trace monoids. Pieces (rectangular solid blocks, or finite unions of such) are stacked in the order of occurrence and mutual dependencies of transitions in some concurrent execution, given by a trace. The height of the stack obtained gives the total makespan under earliest execution. For a number of applications, in particular jobshops, results on asymptotic properties are given in [12, 13]; the probabilistic model for *selecting* the traces is, however, not Markovian.

Finding adequate probability measures for partially ordered, parallel executions of untimed PNs in *logical* time is the subject of [5, 18, 25]. In [5, 25], routing probabilities are used together with the *branching process* semantics from [22, 10], to obtain measures on progressing parallel executions *(configurations)*, filtered by the family of branching prefixes which serve as "temporal" parameters. Limitations of this approach - in both of its variants - did not allow to randomize coherently *arbitrary* net topologies; the indirect dependencies via conflicts and progress of time created situations where renormalization failed.

In [18], we took a *conflict cluster*-centered approach that lifted these restrictions; important aspects will be discussed below, as the approach here runs on a cluster view as well. We obtained a strong Markov property in logical time in [18] (an analogous property is shown in [5]), with respect to stopping times that are particular prefixes of the process net structure; the result suggest that, once physical time is introduced, a regenerative Markov process can be obtained. This is addressed in the present article; it introduces a new, non-branching process semantics that works under the same *cluster view* as [18], but is adapted to the timed framework.

Note that physical time is continuous here, yet it is not 1-dimensional. Keep track of clock states in different parts of the system, height matrix is associated with each logical state (representing a reachable marking of the net); concatenation of processes then translates into $(\max, +)$-multiplication of the associated matrices, as in Winkowski [27, 28, 29].

The paper is organized as follows. Section 2 reviews Petri nets and their dynamics; Section 3 introduces a new process semantics for T-Timed nets that constitutes our first contribution. In Section 4, we exhibit a distributed semi-Markov property satisfied by the STPNs processes introduced here; Section 5 concludes with some final remarks.

2 Definitions

$\mathbb{N}_0$ denotes the set of *non-negative* integers, and $\mathbb{N}$ that of the *positive* integers; $\mathbb{R}$ is the set of real numbers. The $(max, +)$-*semiring* $\mathbb{R}_{max}$ is the set $\mathbb{R} \cup \{-\infty\}$ equipped with the operations max, written $\oplus$, and the usual sum "+", written $\otimes$. Denote $\mathbb{O} \triangleq -\infty$ and $\mathbb{1} \triangleq 0$; then for all $a \in \mathbb{R}_{max}$, $a \oplus \mathbb{O} = \mathbb{O} \oplus a = a$ and $a \otimes \mathbb{1} = \mathbb{1} \otimes a = a$ (see [3, 15] for reference on $(\max, +)$-algebra). Note that $\mathbb{R}_{max}$ is *idempotent,* i.e. $a \oplus a = a$. For matrices $A \in \mathbb{R}_{max}^{n \times m}$ and $B \in \mathbb{R}_{max}^{m \times k}$ and scalar a,

$$\begin{aligned} (A \oplus B)_{ij} &\triangleq A_{ij} \oplus B_{ij} \\ (AB)_{ij} \triangleq (A \otimes B)_{ij} &\triangleq \bigoplus_k A_{ik} \otimes B_{kj} \\ (aA)_{ij} \triangleq (a \otimes A)_{ij} &\triangleq a \otimes A_{ij}. \end{aligned}$$

We denote the vector of dimension $n = |\mathcal{X}|$, for some finite set $\mathcal{X}$, with all entries equal to $\mathbb{1}$ (all entries equal to $\mathbb{O}$) as $\mathbb{1}_n$ ($\mathbb{O}_n$).

Petri Nets. A tuple of sets $N = (\mathcal{P}, \mathcal{T}, F)$ is called a *net* iff $\mathcal{P} \cap \mathcal{T} = \emptyset$, and $F \subseteq [(\mathcal{P} \times \mathcal{T}) \cup (\mathcal{T} \times \mathcal{P})]$. Graphically, F will be indicated by "$\rightarrow$" in the figures; we shall also write $x \rightarrow y$ in the text, if $(x, y) \in F$. The elements of $\mathcal{P}$ will be called *places*, those of $\mathcal{T}$ *transitions*; as usual, we show places as circles and transitions as rectangles. $N' = (\mathcal{P}', \mathcal{T}', F')$ is a *subnet* of N iff **(i)** $\mathcal{P}' \subseteq \mathcal{P}$, **(ii)** $\mathcal{T}' \subseteq \mathcal{T}$, and **(iii)** $F' = F \cap [(\mathcal{P}' \times \mathcal{T}') \cup (\mathcal{T}' \times \mathcal{P}')]$; for $\mathcal{A} \subseteq (\mathcal{B} \cup \mathcal{E})$, the subnet of N spanned by $\mathcal{A}$ is denoted $N[\mathcal{A}]$. A *marking* of N is a multi-set $M : \mathcal{P} \rightarrow \mathbb{N}_0$ of places; if $M(p) = k$, we say there are k *tokens* on place p; tokens are shown as black dots in the figures. A *Petri net* is a tuple $(N, \mathcal{W}, M_0)$, where $N = (\mathcal{P}, \mathcal{T}, F)$ is a finite net, $\mathcal{W} : F \rightarrow \mathbb{N}$ is the *arc weight function*, and $M_0 : \mathcal{P} \rightarrow \mathbb{N}_0$ is the *initial marking*. For a node $x \in (\mathcal{P} \cup \mathcal{T})$, set ${}^\bullet x \triangleq F^{-1}[x]$, $x^\bullet \triangleq F[x]$, and ${}^\bullet x^\bullet \triangleq {}^\bullet x \cup \{x\} \cup x^\bullet$. These notations carry over to *sets* of nodes: if $\mathcal{X} \subseteq \mathcal{P} \cup \mathcal{T}$, then ${}^\bullet \mathcal{X} \triangleq \bigcup_{x \in \mathcal{X}} {}^\bullet x$, $\mathcal{X}^\bullet \triangleq \bigcup_{x \in \mathcal{X}} x^\bullet$, and ${}^\bullet \mathcal{X}^\bullet \triangleq {}^\bullet \mathcal{X} \cup \mathcal{X} \cup \mathcal{X}^\bullet$. If $\mathcal{W}$'s only values are 0 and 1, the Petri net is called *ordinary*.

Dynamics. Transitions may *fire* one by one, or in multi-sets; however, we will assume without loss of generality that the *auto-concurrency degree* is 1, that is, no transition may be in the process of firing more than once at any given time. In fact, any *bounded* net can be modified in such a way that this condition is fulfilled: let $\mathcal{N}$ be k-bounded, and there exists a reachable state of $\mathcal{N}$ in which transition t is firing $m \leqslant k$ times. Then replace t by m copies of itself, with identical pre-and postset, and the same name label t; to each copy t_i, add a place p_i such that $M_0(p_i) = 1$ and ${}^\bullet p_i = p_i^\bullet = t_i$. Then each t_i has auto-concurrency degree 1, and the modified net $\mathcal{N}'$ has the same behaviour as $\mathcal{N}'$ if we count only the number of t_i-labeled transitions firing at any time, and do not distinguish their identities. Hence,

for bounded nets it means no loss of generality when we consider only transition *sets* $\sigma \subseteq \mathcal{T}$ as enabled, possible *steps*; this will be the case throughout the paper. Denote as ε the *empty step*, i.e. $\varepsilon = \emptyset$. A step $\sigma \subset \mathcal{T}$ is *enabled* in a marking M, denoted $M \xrightarrow{\sigma}$, iff M has enough tokens on all $p \in \mathcal{P}$ to satisfy the sum of demands from σ concerning p:

$$\forall\, p \in \mathcal{P}:\ M(p) \ \geqslant \sum_{t \in \sigma \cap p^\bullet} \mathcal{W}(p,t)\,. \tag{1}$$

Denote the set of steps enabled in a marking M as $Enab(M) \triangleq \{\sigma \mid M \xrightarrow{\sigma}\}$. Of course, $\varepsilon \in Enab(M)$ for *any* marking M. Step σ *transforms* marking M into marking M' , denoted $M \xrightarrow{\sigma} M'$, iff *(i)* $M \xrightarrow{\sigma}$, and *(ii)* for all $p \in \mathcal{P}$:

$$M'(p) \ = \ \left[M(p) - \sum_{t \in \sigma \cap p^\bullet} \mathcal{W}(p,t)\right] + \sum_{t \in \sigma \cap {}^\bullet p} \mathcal{W}(p,t)\,. \tag{2}$$

Again, we trivially have $M \xrightarrow{\varepsilon} M$ for *any* marking M. A marking M is *reachable* from M_0, denoted $M_0 \xrightarrow{*} M$, iff: (i) $M = M_0$, or (ii) there exists a *firing sequence* $M_0 \xrightarrow{\sigma_1} M_1 \xrightarrow{\sigma_2} \ldots \xrightarrow{\sigma_n} M_{n+1} = M$.

Stochastic $\mathcal{T}$-timed Petri net (STPNs). Following the terminology of David and Alla [8], an STPN is a tuple of the form $\mathcal{N} = (\mathcal{P}, \mathcal{T}, \mathcal{W}, M_0, \delta)$, where $(\mathcal{P}, \mathcal{T}, \mathcal{W}, M_0)$ is a Petri net, and $\delta = (\delta_t)_{t \in \mathcal{T}}$ a vector of $\mathbb{R}_0^+$- valued random variables. A realization of δ_t gives the duration of a firing instance of t; we assume that the different realizations of δ_t are i.i.d., almost surely finite, and that durations of different transitions are independent. We require, in T-timed nets, the *earliest firing rule*: that is, a transition t that becomes enabled either *(i)* fires immediately itself, or *(ii)* looses concession (immediately) by the firing of a competitor that consumes a token required by t. In no case may t idle in the enabled state for any non-zero amount of time. Then, the firing of a transition t consists of three phases:

1. Removal of "old" tokens from the input places;
2. Firing, which may take time, and during which neither the old nor the new tokens are available; and
3. production of "new" tokens on the output places of t.

Note that the timing from δ acts only in Phase 2, thus influences 3, but cannot resolve conflicts at starting time: contrary to the race policy, the transition timing does not select the transitions to be fired. A probabilistic choice will therefore be assumed for Phase 1, that is, the step to be fired will be chosen in an i.i.d. way (in a sense that will be made precise below). The duration of Phase 2 will obey some law $\mathcal{L}$ that is typically continuous; however, we will not require any particular form of its distribution. It will be necessary to represent the current logical state of an (S)TPN not by its marking alone, but rather a pair

$$\mathcal{S} \ = \ (M, \Theta)\,,$$

where $\Theta \subseteq \mathcal{T}$ is the set of transitions currently firing; recall that we assume auto-concurrency degree 1. Note that $\mathcal{S}$ does *not* indicate the time that a given transition has already been firing.

Causal Nets. The semantics domain here has been in use for a long time in the context of *untimed* nets, see [6]; our extension to T-timed nets follows similar lines as those of [12, 13] and [27, 28, 29]. The structure representing a net's behaviour has itself the form of a net, from the restricted class of *causal nets*:

Let $N = (\mathcal{B}, \mathcal{E}, F)$ a net (not necessarily finite), where we will call the elements of $\mathcal{B}$ the *conditions* and those of $\mathcal{E}$ the *events* of N.

Definition 1. *A net $N = (\mathcal{B}, \mathcal{E}, F)$ is called a* causal net *iff it satisfies:*

1. no branching conditions: $|{}^\bullet b| \leqslant 1$ *and* $|b^\bullet| \leqslant 1$ *for all* $b \in \mathcal{B}$;
2. Acyclicity*: With "$<$" obtained from F as above,* $\neg(x < x)$ *for all* $x \in (\mathcal{B} \cup \mathcal{E})$;
3. N *is* condition-initialized, *i.e.* $\mathbf{c}_0 \triangleq \min_{\leqslant}(\mathcal{B} \cup \mathcal{E}) \subseteq \mathcal{B}$;
4. well-foundedness*: no infinitely $<$-decreasing sequence exists; and*
5. condition-bordered*:* $\max_{\leqslant}(\mathcal{B} \cup \mathcal{E}) \subseteq \mathcal{B}$.

The condition-bordered requirement is non-standard but means no loss of generality; any net meeting all other requirements can be extended into a condition-bordered one, without changing its other properties. From acyclicity, it follows that $<$ is a partial order; the *concurrency relation co* is given by

$$x \; co \; y \quad :\Longleftrightarrow \quad x \neq y \; \wedge \; x \not< y \; \wedge \; y \not< x.$$

Cuts. A $co-$clique $\mathcal{X} \subseteq \mathcal{B}$ of conditions is called a *co-set*. The maximal cliques of co (w.r.t. set inclusion) are called *cuts*; denote the set of cuts as $Cuts(N)$. The cuts consisting only of conditions, i.e. maximal co-sets, will be called *condition-cuts*; the set of condition-cuts is denoted $Cuts_{\mathcal{B}}(N)$. In particular, $\mathbf{c}_0 = \min_{\leqslant}(\mathcal{E} \cup \mathcal{B}) = \min_{\leqslant}(\mathcal{B})$ is a condition-cut. Cuts will represent global states of the net.

Configurations and Runs. For any set $\mathcal{X}$ of nodes of N, call $hull(\mathcal{X}) \triangleq \mathcal{X} \cup {}^\bullet(\mathcal{X} \cap \mathcal{E})^\bullet$ the *condition-bordered* or *open hull* of $\mathcal{X}$. For a node x,let s $x^\downarrow \triangleq \{y \mid y \leqslant x\} \cup \mathbf{c}_0$ be the *past* of x, and

$$\lceil x \rceil \quad \triangleq \quad hull(x^\downarrow)$$

the *condition-bordered local configuration* of x. In general, a *configuration* of N is any node set $\kappa \subseteq (\mathcal{B} \cup \mathcal{E})$ that contains $\mathbf{c}_0$ and such that $x \in \kappa$ implies $\lceil x \rceil \subseteq \kappa$. By abuse of terminology, the subnet $N[\kappa]$ will also be called a configuration. Denote the set of N's configurations as $Conf(N)$. If $\kappa, \kappa' \in Conf(N)$ such that $\kappa \subseteq \kappa'$ when regarded as sets, we say that κ is a *prefix* of κ', written $\kappa \sqsubseteq \kappa'$; it is known that $(Conf, \sqsubseteq)$ is a complete lattice. For all finite configurations κ, let $\mathbf{c}(\kappa)$ be the *bounding cut* of κ, i.e. the set of $<$-maximal conditions of κ.

Height operators. Let $\lambda : \mathcal{B} \cup \mathcal{E} \to [0, \infty)$ be any mapping. We will use λ to represent firing durations of transitions, see below. For $x \in \mathcal{B} \cup \mathcal{E}$, the λ-*height* $\mathbf{h}(b) = \mathbf{h}_\lambda(b)$ of x is given recursively by:

$$\mathbf{h}(x) \quad \triangleq \quad \lambda(x) \otimes \bigoplus_{y \in {}^\bullet x} \mathbf{h}(y) = \lambda(x) + \max\{\mathbf{h}(y) \mid y \in {}^\bullet x\},$$

with the convention $\max(\emptyset) \triangleq \mathbb{1} = 0$. Let κ, κ' be finite configurations $\kappa \sqsubseteq \kappa'$, and write $\mathbf{c} \triangleq \mathbf{c}(\kappa)$ and $\mathbf{c}' \triangleq \mathbf{c}(\kappa')$; let $n \triangleq |\mathbf{c}|$ and $n' \triangleq |\mathbf{c}'|$. The *height matrix* $\mathcal{H}(\mathbf{c}, \mathbf{c}') = \mathcal{H}(\kappa, \kappa') \in \mathbb{R}_{max}^{n \times n'}$ is given by

$$\mathcal{H}_{b,b'} \quad \triangleq \quad \mathbf{d}(b, b') \tag{3}$$

for $b \in \mathbf{c}$ and $b' \in \mathbf{c}'$, where the pseudo-metric $\mathbf{d}$ is defined recursively by

$$\mathbf{d}(x, y) \quad \triangleq \quad \begin{cases} \mathbb{0} & : \; x \; co \; y \\ \bigoplus_{z \in {}^\bullet y} \mathbf{d}(x, z) & : \; x \; < \; y \\ \mathbf{d}(y, x) & : \; y \; < \; x \end{cases} \tag{4}$$

In particular, write $\mathcal{H}_{\mathbf{c}}$ for $\mathcal{H}(\mathbf{c}_0, \mathbf{c})$ and $\mathcal{H}_\kappa$ for $\mathcal{H}(\mathbf{c}_0, \kappa)$. We note that, if $\mathcal{H}$ is given, $\mathbf{h}$ is easily obtained using (3).

Note that the value $\mathcal{H}_{b,b'}$ indicates the length, under λ, of the longest path leading from b to b' in configuration κ'. $\mathcal{H}_\kappa$ as above can also be seen as a $(\max, +)$-linear operator:

$$\mathbf{h}_\kappa : \mathbb{R}_{max}^{n_0} \quad \to \quad \mathbb{R}_{max}^{n} \tag{5}$$

$$x_0 \quad \mapsto \quad x_\kappa \triangleq \mathbf{h}_\kappa \otimes x_0, \tag{6}$$

In our setting, $x_0 \; \mathbb{1}_{n_0}$.

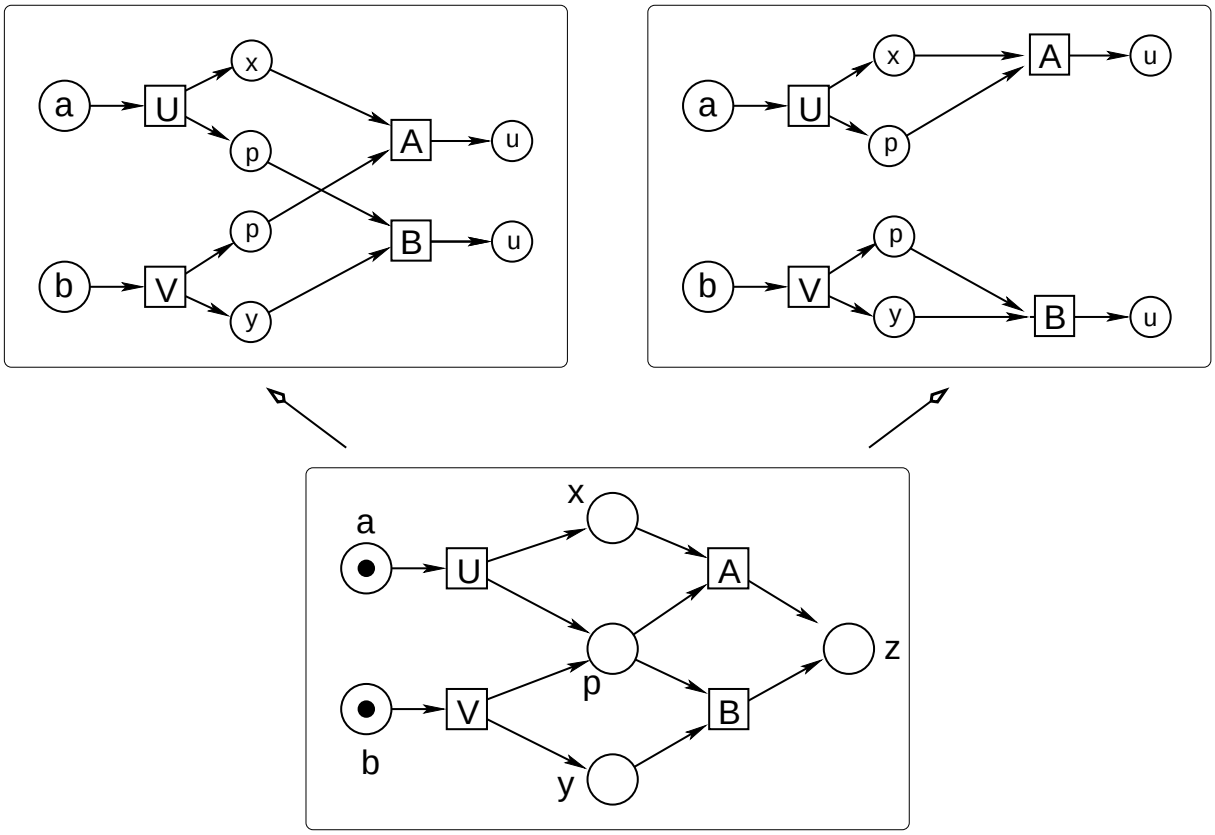

Figure 1: Process semantics under the individual token view. Below the Petri net whose maximal concurrent processes are shown above.

Concatenation.

Definition 2. *Let $N = (\mathcal{B}, \mathcal{E}, F)$ be a finite causal net with final cut $\mathbf{c}$, and let $N' = (\mathcal{B}', \mathcal{E}', F')$ be a causal net such that there is a bijection ϕ between the initial cut $\mathbf{c}'_0$ of N' and $\mathbf{c}$. The causal net $N \circ N' = (\mathcal{B}'', \mathcal{E}'', F)$ with*

$$\begin{aligned} \mathcal{E}'' &\triangleq \mathcal{E} \cup \mathcal{E}', \\ \mathcal{B}'' &\triangleq (\mathcal{B} \backslash \mathbf{c}) \cup \mathcal{B}', \\ F'' &\triangleq F \cup F' \cup \{(e, \phi(b)) \mid (e, b) \in F\}, \end{aligned}$$

is called the concatenation *of N' and $\mathbf{c}$.*

The property most relevant here is the following:

Lemma 1. *In the situation of Def. 2, with N' finite and $\mathbf{c}'$ its final cut, we have:*

$$\mathcal{H}_{N \circ N'} = \mathcal{H}_{N'} \otimes \mathcal{H}_N. \tag{7}$$

Seen as a relation of operators following (5), (1) becomes

$$x_{N \circ N'} = \mathcal{H}_{N \circ N'} \otimes x_0 = \mathcal{H}_{N'} \otimes x_N = \mathcal{H}_{N'} \otimes \mathcal{H}_N \otimes x_0. \tag{8}$$

The proof is straightforward from the definitions.

3 Processes

Process semantics for Petri nets associate, to a given $\mathcal{N}$, causal nets with mappings that reflect the arc structure and the flow of tokens. In contrast to other occurrence net semantics, the one we will introduce below will contain conditions that represent the duration of a firing of some transition t, and are therefore not mapped to any place of $\mathcal{N}$.

Both Gaubert/Mairesse [12, 13] and Winkowski [27, 28, 29] used matrix representation of Petri net processes to obtain a $(\max, +)$-representation of timed behavior. The main difference is that Winkowski does not require 1-boundedness; still, the semantics in his approach is the standard non-sequential process semantics under the individual token view, of which a comprehensive treatment can be found in [6]. Figure 1 shows a small Petri net with its two maximal "token view" processes. Each token presence in a place gives rise to a condition in the causal nets above; thus, since place p receives two tokens during the execution, it is represented twice. The two processes shown are non-isomorphic, yet their physical difference lies only in the history of tokens, i.e. which transition produced the token on p consumed by A. However, this distinction of token individualities is not natural, but rather a particularity of the semantics used to observe it; to free the computations of the resulting combinatorial burden, Winkowski [27, 28, 29] performs an additional step to abstract away from the permutations of tokens in a place.

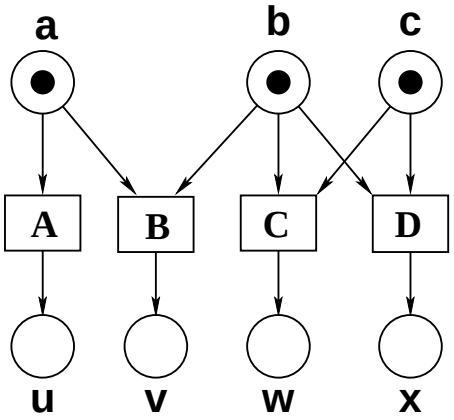

Figure 2: Conflicts in clusters

Here, we will treat places as variables, and the number of tokens currently present on place p as the current value of variable p; transitions act on these variables, adding or subtracting from their values. This *collective token view* abstracts consistently from token individuality; it is at the heart of *execution* semantics, see [26, 11, 16]. However, that semantics has two drawbacks:

1. it allows for no parallel firing of different transitions sharing a common place, even if the marking provides enough tokens for simultaneous firings; and
2. the indirect dependency of transitions engaged in different conflicts are not expressible: in Figure 2, transitions A and C are not in conflict; yet the probability of A firing is changed by information about C. If C fires, A is without competition, and under the earliest firing rule, its firing is assured.

For those reasons, we used a modified view in [17] and [18]:

1. Conflicts are resolved not by a single transition or place, but jointly by the entire *conflict cluster* (called *in-cluster* here, see below).
2. Consequently, events do not represent single firings of transitions in general, but rather occurrences of *steps*.

Clusters. $\mathcal{N}$ is naturally partitioned into node sets that are minimally closed under conflicts. Note that *"conflict"* can be taken in the forward or backward sense, since places may be forward and/or backward branching; hence we distinguish to types of clusters:

Definition 3. *The* (in-)cluster $\mathsf{inc} = \mathsf{inc}(x)$ *of* $x \in (\mathcal{P} \cup \mathcal{T})$ *is the smallest set containing* x *that satisfies:*

$$\forall t \in \mathcal{T} : \; t \in \mathsf{inc} \Rightarrow {}^\bullet t \subseteq \mathsf{inc} \quad \textit{and} \quad \forall\, p \in \mathcal{P} : \; p \in \mathsf{inc} \Rightarrow p^\bullet \subseteq \mathsf{inc}.$$

By extension, call cluster *any set* $\mathsf{inc} \subseteq (\mathcal{P} \cup \mathcal{T})$ *for which there exists a node* x *such that* $\mathsf{inc} = \mathsf{inc}(x)$*; denote the set of clusters of* $\mathcal{N}$ *as* $\mathsf{InC}(\mathcal{N})$*. Dually, the* out-cluster *of node* x *is the smallest set* $\mathsf{outc} \subseteq (\mathcal{P} \cup \mathcal{T})$ *containing* x *that satisfies:*

$$\forall t \in \mathcal{T} : \; t \in \mathsf{outc} \Rightarrow t^\bullet \subseteq \mathsf{outc} \quad \textit{and} \quad \forall\, p \in \mathcal{P} : \; p \in \mathsf{outc} \Rightarrow {}^\bullet p \subseteq \mathsf{outc},$$

and $\mathsf{OutC} = \mathsf{OutC}(\mathcal{N})$ *denotes the set of out-clusters of* $\mathcal{N}$*. For* $\mathsf{inc} \in \mathsf{InC}$ *and* $\mathsf{outc} \in \mathsf{OutC}$*, abbreviate*

$$\mathcal{P}_{\mathsf{inc}} \triangleq \mathcal{P} \cap \mathsf{inc}, \; \mathcal{P}_{\mathsf{outc}} \triangleq \mathcal{P} \cap \mathsf{outc} \textit{ and } \mathcal{T}_{\mathsf{inc}} \triangleq \mathcal{T} \cap \mathsf{inc}, \; \mathcal{T}_{\mathsf{outc}} \triangleq \mathcal{T} \cap \mathsf{outc}.$$

See Figure 3 for an illustration.

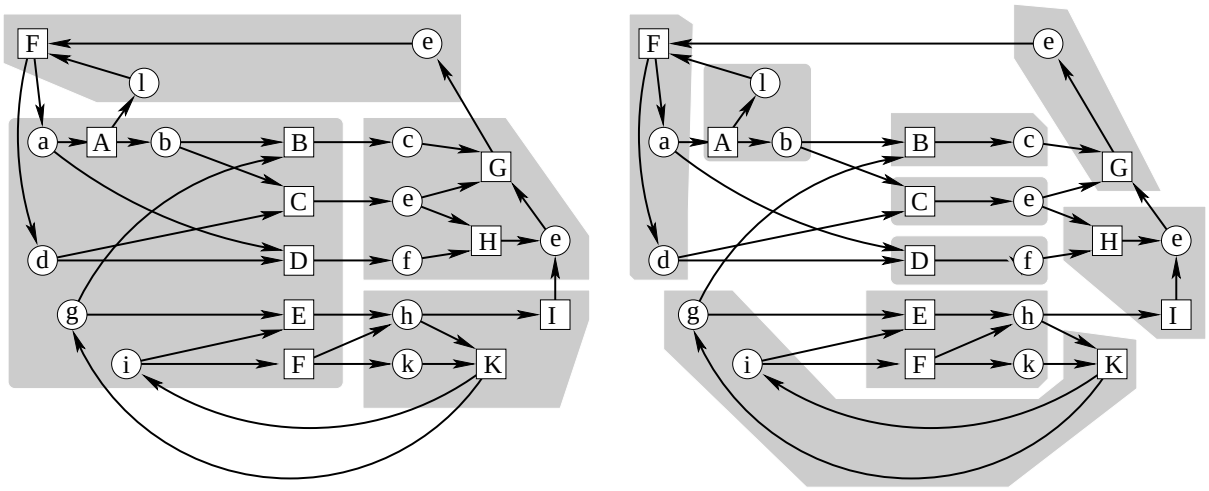

Figure 3: A net decomposed into its clusters (left) and out-clusters (right)

Remark 1. *In-Clusters are standard in the literature, mostly under the name of* conflict clusters*, see [9]. The stochastic processes in logical distributed time studied in [18] are based on local choices (for transition firings to be initiated) of clusters, scheduled by a distributed control mechanism; out-clusters were not needed. Here, we need to consider out-clusters to observe the order in which different transitions, upon completion of their firing, put on the same receiving place; the duality of in- and out-clusters reflects that of beginning and end of a timed transition firing. The algorithm below will reflect that distribution of roles: in-clusters select the transitions to be started, out-clusters coordinate the endings of firings.*

Figure 4 shows two clusters (plus their output places), with the first stage of the process construction: both select some firable step (not all combinations are shown), create the corresponding events, change the value of input places, and append the duration conditions (shaded in grey) for the transitions in the selection. The *end*-events, marking the end of a firing instance that produces tokens, are not produced in this stage; that phase is driven by the out-clusters, as shown in Figure 5 and explained in the following algorithm.

The choice of a step in a cluster can be done according to any measure concentrating its mass on *maximal* steps (w.r.t. step inclusion); all other firings either violate the earliest firing rule, or increase the

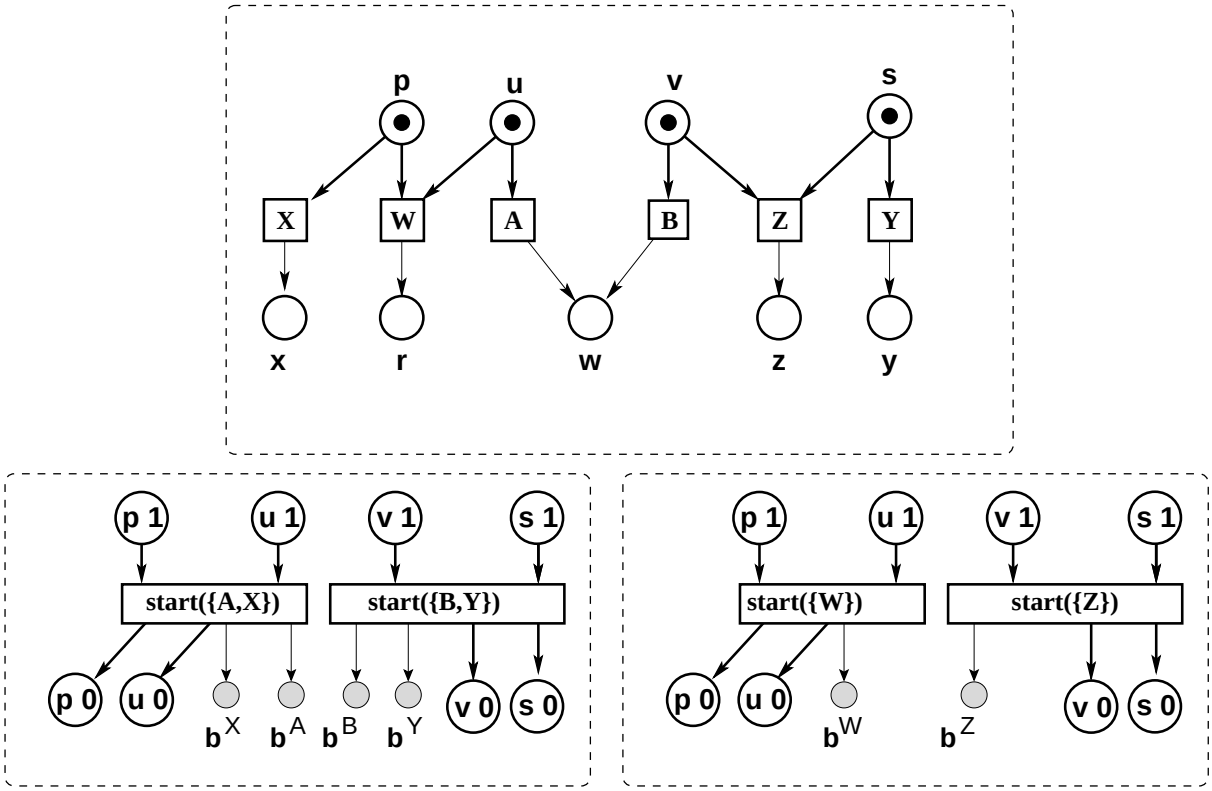

Figure 4: Process semantics under the cluster view.

number of nodes in the configuration without necessity. such probability measures may be constructed, as in [18], using a *Gibbs potential* approach that maximizes conditional independences between transitions of the same cluster; however, this is not essential here. More importantly, we assume that all choices are made in an independent way, see below.

The principles of the process semantics. Compared to the well-known semantics discussed above, some modifications are necessary to incorporate timing. The fact that transition t is firing, will be reflected by a dedicated condition b^t. Condition b^e will connect the events marking the beginning and end of the firing, respectively. Contrary to other conditions, b^e does not correspond to any place of the net; the vale $\lambda(b^t)$ is to be interpreted as the *duration* of the firing instance e of t. All other conditions correspond to logical states only, and will not be assigned a duration, i.e. λ takes value 0 for those conditions. Also, λ is set to 0 for *all* events: in the semantics we exhibit here, events represent beginnings or endings of firings, and therefore serve merely to separate time intervals; they have no duration of their own.

Algorithmic construction of processes. The main ingredients will be a family of causal nets, and a family of mappings of three different types:

1. associating *conditions* to *places*; these mappings will be denoted π;
2. associating *events* to *steps*, i.e. *sets of transitions* , and denoted ρ;
3. and finally, mappings denoted μ and giving *token loads* to conditions.

In the figures, we include the value of μ in the labels on conditions, e.g. "$a1$" for a condition that is mapped to place a by π, with one token on a: $\mu(a1) = 1$. Further, we assume knowledge of duration

δ, given under the form of a mapping λ that maps to 0 unless its argument is a duration condition; the durations will impact also the *shape* of the process nets.

Algorithm PROC

- *Initialize* $\kappa_0 \triangleq \mathbf{c}_0$, *let* $\pi_0 \triangleq \pi_{\kappa_0} : \mathbf{c}_0 \to \mathcal{P}$ *a bijection, and* $\mu_0 \triangleq \mu_{\mathbf{c}_0} : \mathbf{c}_0 \to \mathbb{N}_0$ *such that* $\mu_0(b) = M_0(\pi_0(b))$ *for all* $b \in \mathbf{c}_0$, *and* $\mathcal{M}_0 = \mathcal{M}_{\kappa_0} \triangleq \mathbb{O}$.
- *Let a configuration* κ, *together with* $\mathbf{c} = \mathbf{c}(\kappa)$, $M(\kappa)$, π_κ, $\mathbf{h}_{|\kappa}$, *and* μ_κ *be given.*
  1. *For every cluster* inc, *make a choice of a* $M(\kappa)$*-enabled step* σ *of* inc*-transitions, and append a new event* $start(\sigma)$ *to* κ *such that*
  $$^\bullet start(\sigma) \quad = \quad \{b \in \mathbf{c}_\kappa \mid \pi_\kappa(b) \in \mathcal{P}_{\mathsf{inc}}\}.$$
  2. *For all* $p \in \pi(^\bullet start(\sigma))$, *create a post-condition* b_p *of* $start(\sigma)$; *set*
  $$\begin{aligned} \pi_{n+1}(b_p) &\triangleq p \\ \mu_{n+1}(b_p) &\triangleq \left[\mu_n(b_p^{in})) - \sum_{t \in {}^\bullet p \cap \sigma} \mathcal{W}(p,t)\right] + \sum_{t \in p^\bullet \cap \sigma} \mathcal{W}(t,p), \end{aligned}$$
  where b_p^{in} *is the unique pre-condition of* e_σ *such that* $\pi_n(b_p) \triangleq p$.
  3. *For each* $t \in \sigma$, *let* b^t *be a new post-condition of* $start(\sigma)$. *Let* $\lambda(b^t) \in [0, \infty)$ *be the duration of the firing event of* t *in* σ.
- *Set* $\mathcal{X} \triangleq \mathsf{OutC}$, *and repeat until* $\mathcal{X} = \emptyset$*:*
  1. *Take* $\mathsf{outc} \in \mathcal{X}$; *for every* $t \in \sigma \cap \mathcal{T}_{\mathsf{outc}}$, *add an event* $end(t)$. *Set*
  $$\mathbf{h}(e) \quad \triangleq \quad \mathbf{h}(b) \otimes \lambda(b^t),$$
  and let
  $$\begin{aligned} end(t) \prec end(t') \quad &\textit{iff} \quad \left[\mathbf{h}\left(end(t)\right) \leqslant \mathbf{h}\left(end(t')\right) \ \wedge \ t^\bullet \cap (t')^\bullet \neq \emptyset\right] \\ end(t) \sim end(t') \quad &\textit{iff} \quad \left[\mathbf{h}\left(end(t)\right) = \mathbf{h}\left(end(t')\right) \ \wedge \ t^\bullet \cap (t')^\bullet \neq \emptyset\right]. \end{aligned}$$
  2. *Let* m *be a* $\sim$*-class of* $end(\bullet)$*-events; contract all events in* m *into a single event* $end(m)$ *(compare Figure 5).*
  3. *Let* $End(\sigma) = \{end_1, \ldots, end_c\}$ *be the set of* $end(\bullet)$*-events, after reduction, for* σ, *and assume* $end_1 \prec \ldots \prec end_s$. *For all places* $p \in \mathcal{P}_{\mathsf{outc}}$, *add new conditions* $b_1, \ldots, b_s$ *such that*
     (a) $b_0 \to end_1$, *where* b_0 *is the unique* p*-labeled condition in* $start(\sigma)^\bullet$, *and*
     (b) $end_i \to b_i \to endi + 1$ *for* $1 \leqslant i \leqslant s - 1$.
     Split each of these conditions b_i *in its into two conditions* $\overline{b}_i$ *and* $\underline{b}_i$, *such that*
     $$end_i \to \underline{b}_i \to start(\sigma') \to \overline{b}_i \to end_{i+1},$$
     where σ' *is a step of* b_i*'s own in-cluster* $\mathsf{inc}(b_i)$ *that became enabled by the new tokens created by the completion of some* t*'s firing. Each condition* b_i *in the above can undergo several such subdivisions, since the in-cluster* $\mathsf{inc}(\pi(b_i))$ *receives tokens from several transitions, and may thus have new enabled steps. Since* $\pi(b_i)$ *belongs to only one cluster, only a finite number of splittings can occur because of the earliest firing rule.*

Denote the new configuration thus obtained as κ', set $\lambda(x) \triangleq 0$ for all new nodes EXCEPT the duration conditions b_σ, and repeat, with κ replaced by κ' and $\mathcal{X} \triangleq \mathcal{X}\backslash\{\mathsf{outc}\}$, until $\mathcal{X} = \emptyset$.

Denote as κ^+ the configuration obtained after termination of this algorithm. Note that different (out-)clusters act on disjoint sets of places. Therefore, κ^+ is well-defined up to isomorphism, i.e. does not depend on the order in which the $\mathsf{outc} \in \mathsf{OutC}$ are considered by the algorithm.

The tuple $\Pi = (\kappa, \pi, \rho, \mu, \mathcal{H})$ thus obtained is called a *process* of $\mathcal{N}$. It is a random element, determined jointly by *(i)* the choice of enabled steps in a marking reached, and *(ii)* the random firing duration of transitions, sampled from δ. Note that these components can not be separated in general, since the choices available along a process will be influenced also by the timed behavior.

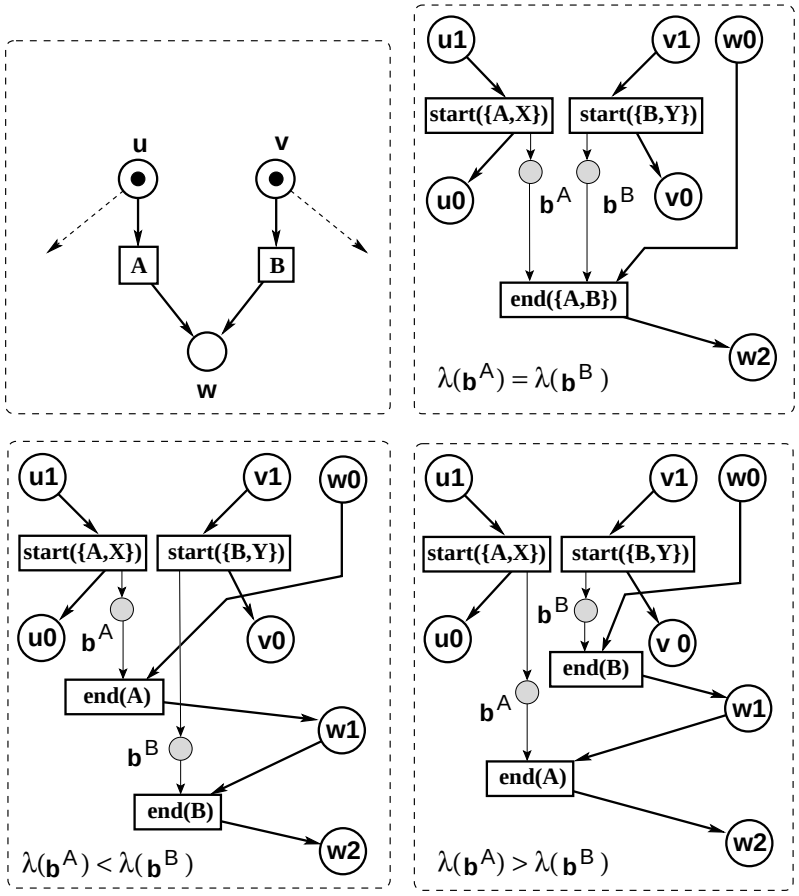

Figure 5: Temporal Orderings and associated process structures.

Regular configurations. Considering the configurations that are contained as prefixes in a process generated by **PROC**, one observes that some of those configurations contain duration conditions b^e in their final cut, while others do not. For κ of the latter type, we have that

$$\mathbf{c}(\kappa) \text{ is in bijection with } \mathcal{P} \text{ via } \pi. \tag{9}$$

Call all configurations of Π such that (9) holds, and their final cuts, **regular**. One notices that regular configurations lead, in general, to transient states (more precisely, because of the earliest firing rule, a regular cut corresponds to a non-transient state only if it is dead; in that case, the configuration has no continuation). Observable states correspond, generally, to non-regular cuts, which contain some duration conditions that indicate a firing in process. Despite this fact, the configurations and cuts of the first type,

which we will call *regular,* correspond to the regeneration states, and form the Markov chain inscribed in the Semi-Markov Process we will exhibit below.

It is important to note that the configuration $\kappa+$ obtained from κ by Algorithm **PROC** is again regular. Since the initial configuration consisting only of $\mathbf{c}_0$ is regular, we may assume that **PROC** is applied only to regular configurations.

Concatenation and suffixes. Let $\Pi = (\kappa, \pi, \rho, \mu, \mathcal{H}_\kappa)$ be a finite process of $\mathcal{N} = (\mathcal{P}, \mathcal{T}, \mathcal{W}, \delta, F, M_0)$, let M_Π the marking reached after Π, i.e. the marking corresponding to $\mathbf{c} \triangleq \mathbf{c}(\kappa)$, and

$$\Theta(\kappa) \quad \triangleq \quad \left\{ t \in \mathcal{T} \mid b^t \in \mathbf{c} \right\}$$

the set of transitions currently firing at the end of κ. Then $\mathcal{S}(\Pi) \triangleq (M(\kappa), \Theta(\kappa))$ is the *state* of $\mathcal{N}$ after Π. We first note that $\mathcal{H}(\kappa, \kappa')$ is a square matrix for all *regular* configurations κ' such that $\kappa \sqsubseteq \kappa'$; this is *not* true for general configurations, of course. Let $\Pi' = (\kappa', \pi', \rho', \mu', \mathcal{H}_{\kappa'})$ be another process of $\mathcal{N}$. We say that Π is a *prefix* of Π', written $\Pi \sqsubseteq \Pi'$, iff $\kappa \sqsubseteq \kappa'$, and π, ρ, μ are restrictions of π', ρ', μ' to κ. If $\Pi \sqsubseteq \Pi'$,let $\tilde{\kappa}''$ be the suffix of κ' obtained after cutting κ' at $\mathbf{c}$, and $\kappa'' \triangleq \tilde{\kappa}'' \cup \mathbf{c}$. Then κ'' is concatenable with κ in the sense of Definition 2, using as ϕ the matching of conditions corresponding to the same place in $\mathcal{P}$, and duration conditions corresponding to the same transition. From Lemma 1, we know that

$$\mathcal{H}_{\kappa \circ \kappa''} \quad = \quad \mathcal{H}_\kappa \otimes \mathcal{H}_{\kappa''}; \tag{10}$$

hence, we call Π' the *concatenation* $\Pi \circ \Pi''$ of Π with $\Pi'' \triangleq (\kappa'', \pi'', \rho'', \mu'', \mathcal{H}_{\kappa''})$, where π'', ρ'', μ'' are the restrictions of π', ρ', μ' to κ'. Π'' thus obtained is a *process suffix* of $\mathcal{N}$.

In Figure 5, all heights are given by δ since we start in $\mathbf{c}_0$. The figure continues Figure 4, restricting attention to $\mathsf{outc}(w)$; not all input conditions of $start(\{A, X\})$ and $start(\{B, Y\})$ are represented, b^Y and b^X are dropped, etc. Figure 5 shows that differences in the duration of transition firings can change the structure of the configuration. It is thus **not** possible to first generate the causal net, and then add durations: the timing influences the simultaneous availability or non-availability of tokens, and the earliest firing rule forbids waiting for a missing token to arrive.

In the example of Figure 5 (forgetting the nodes present in Figure 4 but omitted in Figure 5), let κ the maximal configuration in any of the three cases shown, $\mathbf{c}_0$ be $\mathbf{c}_0 = \{u1, v1, w0\}$, and $\mathbf{c}_1 = \mathbf{c}(\kappa) = \{u0, v0, w2\}$. With this ordering, we obtain the height matrix

$$\mathcal{H}_\kappa \quad = \quad \begin{pmatrix} \mathbb{1} & \mathbb{0} & \lambda(b^A) \oplus \lambda(b^B) \\ \mathbb{0} & \mathbb{1} & \lambda(b^A) \oplus \lambda(b^B) \\ \mathbb{0} & \mathbb{0} & \mathbb{1} \end{pmatrix}, \tag{11}$$

or, with $\rho \triangleq \lambda(b^A) \oplus \lambda(b^B)$,

$$\mathcal{H}_\kappa \quad = \quad \begin{pmatrix} \mathbb{1} & \mathbb{0} & \rho \\ \mathbb{0} & \mathbb{1} & \rho \\ \mathbb{0} & \mathbb{0} & \rho \end{pmatrix}. \tag{12}$$

Note that the $\otimes$-operation allows to have a unified form of $\mathcal{H}_\kappa$ for all three cases exhibited in Figure 5.

4 Distributed Semi-Markov processes

Now, let $\delta(t, k)$ be a family of firing times, with t ranging over $\mathcal{T}$ and k over $\mathbb{N}$, such that

(D1) $(t, k) \neq (t', k')$ implies that $\delta(t, k)$ and $\delta(t', k')$ are independent, and

(D2) for fixed transition t, $(\delta(t, k))_{k \in \mathbb{N}}$ is i.i.d.

Similarly, let $\sigma(\mathsf{inc}, M_{\mathsf{inc}}, n)$ be a family of random variables with inc varying over InC, M over the multi-sets over $\mathcal{P}_{\mathsf{inc}}$, and n over $\mathbb{N}$, such that:

(C1) For all clusters inc and markings M_{inc} on the places of inc, and all $n \in \mathbb{N}$, $\sigma(\mathsf{inc}, M_{\mathsf{inc}}, n)$ is almost surely a maximal step enabled in M_{inc};

(C2) for all clusters inc and markings M_{inc} on the places of inc, the sequence $(\sigma(\mathsf{inc}, M_{\mathsf{inc}}, n))_{n\in\mathbb{N}}$ is i.i.d, and

(C3) for any two distinct clusters inc and inc$'$ with local markings M_{inc} and $M_{\mathsf{inc}'}$, and any $n, n' \in \mathbb{N}$, $\sigma(\mathsf{inc}, M_{\mathsf{inc}}, n)$ and $\sigma(\mathsf{inc}', M_{\mathsf{inc}'}, n')$ are independent.

Under these hypotheses, it is not hard to see that STPN $\mathcal{N}$ with the earliest firing rule generates "a sort of" heterogeneous Semi-Markov process; however, to state this properly, we have to leave the standard time model with scalar time parameter. In fact, our time parameter here is matrix-valued in $\mathbb{T} \triangleq \mathbb{R}_{max}^{n\times n}$, where n is the number $n \triangleq |\mathcal{P}|$ of places in $\mathcal{N}$. Let us first note that $\mathbb{T}$ with the natural ordering $\preccurlyeq$,

$$\mathcal{M} \preccurlyeq \mathcal{M}' \quad \textit{iff} \quad \left(1 \leqslant i,j \leqslant n \Rightarrow \mathcal{M}_{i,j} \leqslant \mathcal{M}'_{i,j}\right),$$

is a conditionally complete lattice. For $\mathcal{M}_1 \preccurlyeq \mathcal{M}_2$, denote as

$$[\mathcal{M}_1, \mathcal{M}_2] \quad \triangleq \quad \{\mathcal{M} \mid \mathcal{M}_1 \preccurlyeq \mathcal{M} \preccurlyeq \mathcal{M}_2\}$$

the closed interval bounded by $\mathcal{M}_1$ and $\mathcal{M}_2$.

Shifting the standard definitions, from the setting with time set $\mathbb{R}$ and some discrete state space, to our framework with the lattices $\mathbb{T}$ and $\mathit{Conf}(\mathcal{N})$, we obtain that the process generated by $\mathcal{N}$ via **PROC** satisfies the following, semi-Markovian property:

Theorem 1. *Let $\mathcal{N}$ be an STPN, and assume (D1-2) and (C1-3). Let $\Pi_1, \ldots, \Pi_n$, Π and Π' such that $\Pi_1 \sqsubseteq \ldots \sqsubseteq \Pi_n = \Pi \sqsubseteq \Pi'$ be processes of $\mathcal{N}$, with Π' the random element whose conditional law is wanted.Further, fix some markings M, M_i, transition sets $\Theta, \Theta_i \subseteq \mathcal{T}$; set $\mathcal{S} \triangleq (M, \Theta)$ and $\mathcal{S}_i \triangleq (M_i, \Theta_i)$, and let $\mathcal{M}$, $\mathcal{M}_i$ for $1 \leqslant i \leqslant n$ be fixed matrices of appropriate dimensions, such that the products in (13) are defined. Then:*

$$\begin{aligned} & \mathbb{P}\left(\begin{array}{l} \mathcal{S}(\kappa') = \mathcal{S} \\ \mathcal{M}_{\kappa'} \in [\mathcal{M}_\kappa, \mathcal{M}_\kappa \otimes \mathcal{M}] \end{array} \middle| \forall\, i:\ (\mathcal{S}(\kappa_i), \mathcal{M}_{\kappa_i}) = (\mathcal{S}_i, \mathcal{M}_i) \right) \\ = \quad & \mathbb{P}\left(\begin{array}{l} \mathcal{S}(\kappa') = \mathcal{S} \\ \mathcal{M}_{\kappa'} \in [\mathcal{M}_\kappa, \mathcal{M}_\kappa \otimes \mathcal{M}] \end{array} \middle| (\mathcal{S}(\kappa_n), \mathcal{M}_{\kappa_n}) = (\mathcal{S}_n, \mathcal{M}_n) \right). \end{aligned} \tag{13}$$

Proof: By inspection of Algorithm **PROC** above. □

5 Final Remarks

We have introduced a process semantics for T-timed Petri nets under the cluster view. Under i.i.d. assumptions on choice and duration distributions, an analogue (13) of the semi-Markov property was seen to hold. In (13), the usual linear time parameter is replaced by matrix-valued multi-dimensional time, taking the form of $(\max, +)$ operators; this is precisely why conditional independence from the past holds, even for duration distributions with memory. The operator lattice $\mathbb{T}$, from which the consumption of physical time can be read, reflects the lattice *Conf* of configurations describing the possible evolutions in logical time.

The above approach extends the representational results of Gaubert and Mairesse [12, 13] and Winkowski [27, 28, 29]. It will allow to carry out, in a first step, asymptotic analysis using non-expansive operator theory; bringing this to work is the subject of work in progress.

References

[1] M. Ajmone Marsan, G. Balbo, G. Conte, S. Donatelli, and G. Franceschinis. *Modeling with Generalized Stochastic Petri Nets.* Parallel Computing Series, Wiley, 1995.

[2] M. Ajmone Marsan, G. Balbo, G. Chiola, and G. Conte. *Generalized Stochastic Petri Nets Revisited: Random Switches and Priorities. Proc. PNPM '87*, IEEE-CS Press, pp. 44–53.

[3] F. Baccelli, G. Cohen, G. J. Olsder, and J.-P. Quadrat. *Synchronization and Linearity.* Wiley, 1992.

[4] F. Baccelli and J. Mairesse. Ergodic theorems for stochastic operators and discrete event networks. In: [15].

[5] A. Benveniste, S. Haar, and E. Fabre. Markov Nets: probabilistic Models for Distributed and Concurrent Systems. *INRIA Report* **4235**, September 2001. Submitted to a journal.

[6] E. Best and C. Fernández. *Nonsequential Processes. A Petri Net View. EATCS Monographs* **13**, Springer Verlag 1988.

[7] P. Cartier and D. Foata. *Problèmes combinatoires de commutations et réarrangements.* Lecture Notes in Math. **85**, Springer, 1969.

[8] R. David and H. Alla. Petri nets for modeling of dynamical systems – a survey. *Automatica* **30**(2):175–202, 1994.

[9] J. Desel and J. Esparza. *Free Choice Petri Nets.* Cambridge University Press, 1995.

[10] J. Engelfriet. Branching Processes of Petri Nets. *Acta Informatica* **28**:575–591, 1991.

[11] J. Esparza, S. Römer, and W. Vogler. An Improvement of McMillan's Unfolding Algorithm. *Formal Methods in System Design* **20**(3):285–310, May 2002.

[12] S. Gaubert and J. Mairesse. Asymptotic Analysis of Heaps of Pieces and application to Timed Petri Nets. *Proc. PNPM '99*, 158–169.

[13] S. Gaubert and J. Mairesse. Modeling Analysis of Timed Petri Nets Using Heaps of Pieces. *IEEE Trans. Aut. Control* **44**(4):683–687, 1999.

[14] B. Gaujal, S. Haar, and J. Mairesse. Blocking a transition in a Free Choice net and what it tells about its throughput. To appear in *Journal of Computer and System Science.* Preliminary version: *INRIA Research Report* **4225**, July 2001; downloadable at URL: http://www.inria.fr/rrrt/rr-4425.html.

[15] J. Gunawardena (ed.), *Idempotency*, Cambridge University Press, 1998, p.171–208.

[16] S. Haar. Branching Processes of general S/T-Systems. *Workshop on Concurrency at MFCS '98, Brno. Electronic Notes in Theoretical Computer Science* **18**, 1998, Elsevier; `http://www.elsevier.nl/locate/entcs/volume18.html`.

[17] S. Haar. Probabilistic Unfoldings and Partial Order Fairness in Petri Nets. In: H. Hermanns and R. Segala (eds.), *Proceedings PAPM-ProbMiV 2002*, LNCS **2399**, 95–114.

[18] S. Haar. Probabilistc Cluster Unfoldings. To appear in *Fundamenta Informaticae.*

[19] P. J. Haas. *Stochastic Petri Nets. Modelling, Stability, Simulation.* Springer Series in Operations Research, Springer Verlag, 2002.

[20] C. Lindemann and G. Shedler. Numerical Analysis of Deterministic and Stochastic Petri Nets with Concurrent Deterministic Transitions. *Performance Evaluation, Special Issue Proc. of PERFORMANCE '96* **27/28**:565-582, 1996.

[21] C. Lindemann and A. Thümmler. Transient Analysis of Deterministic and Stochastic Petri Nets with Concurrent Deterministic Transitions. *Performance Evaluation* **36/37**:35–54, 1999.

[22] M. Nielsen, G. Plotkin, and G. Winskel. Petri nets, event structures, and domains. Part I. *Theoretical Computer Science* **13**:85–108, 1981.

[23] J. Mairesse and L. Vuillon. Asymptotic behavior in a heap model with two pieces. *Theoretical Computer Science* **270**:525–560, 2002.

[24] G.X. Viennot. Heaps of pieces, I: Vasic definitions and combinatorial lemmas. In: Labelle and Leroux (eds.), *Combinatoire Énumérative*, *Lecture Notes in Mathematics* **1234**:321–350, Springer, 1986.

[25] H. Völzer. Randomized non-sequential processes. *CONCUR*. LNCS **2154**, Springer, 2001.

[26] W. Vogler. Executions: A New Partial Order Semantics of Petri Nets. *Theoretical Computer Science* **91**:205-238, 1991.

[27] J. Winkowski. Processes of timed Petri nets. *Theoretical Computer Science* **243**(1-2): 1–34 (2000).

[28] J. Winkowski. A Representation of Processes of Petri Nets by Matrices. *Fundamenta Informaticae*, **30**(1):97–107, 1997.

[29] J. Winkowski. Concatenable weighted pomsets and their applications to modeling processes of Petri Nets. *Fundamenta Informaticae*, **1**:1–6, 1996.

Unité de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unité de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unité de recherche INRIA Rhône-Alpes, 655, avenue de l'Europe, 38330 MONTBONNOT ST MARTIN
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
http://www.inria.fr
ISSN 0249-6399