Probabilistic Cluster Unfoldings for Petri Nets

Stefan Haar

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Probabilistic Cluster Unfoldings for Petri Nets

Stefan Haar

N˚4426

Avril 2002

THÈME 4

Rapport de recherche

ISSN 0249-6399 ISRN INRIA/RR--4426--FR+ENG

Probabilistic Cluster Unfoldings for Petri Nets

Stefan Haar*

Thème 4 — Simulation et optimisation
de systèmes complexes
Projets SIGMA2

Rapport de recherche n ° 4426 — Avril 2002 — 29 pages

Abstract: This article introduces a probabilistic unfolding semantics for untimed Petri nets. No structural or safety assumptions are made. We show that *cluster semantics* is an adequate framework for the construction of probability measures for concurrent runs. The unfolding semantics is constructed by local choices on each cluster, and a distributed scheduling mechanism (*cluster net*) authorizing cluster actions.The probability measures for the choice of step in a cluster are obtained by constructing Markov Fields on the conflict graph of transitions, from suitable Gibbs potentials. We introduce and characterize stopping times for these models, and a strong Markov property.

Key-words: Probabilistic methods, Concurrency, Petri net unfoldings, cluster semantics, distributed systems.

(Résumé : tsvp)

* IRISA/INRIA, Campus de Beaulieu, 35042 Rennes cedex, France. Supported by RNRT through the MAGDA project

Unité de recherche INRIA Rennes

Dépliages probabilistes à Clusters des Réseaux de Pétri

Résumé : Cet article introduit une sémantique probabiliste pour les Réseaux de Pétri non temporisés. Aucune hypothèse n'est exigée concernant la structure ou la sûreté. Nous démontrons que la *sémantique des clusters* donne un cadre adapté à la construction de mesures de probabilité pour les exécutions en parallèle. Le dépliage est construit à partir des choix locaux sur chaque cluster, ainsi qu'un mécanisme distribué (*cluster net*) qui donne l'autorisation d'action aux clusters. Les mesures de probabilité gouvernant le choix d'action dans un cluster sont obtenues en construisant, à partir de potentiels de Gibbs adaptés, des champs de Markov sur le graphe de conflit des transitions. Pour ces modèles, nous introduisons des temps d'arrêt et donnons une caractérisation pour ceux-là, et démontrons une proprieté de Markov forte.

Mots-clé : Probabilité, Parallèlisme, Dépliages, sémantique de clusters, systèmes repartis.

Contents

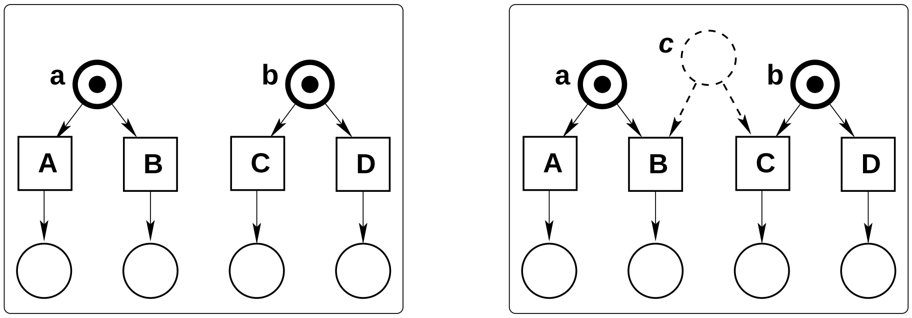

Figure 1: Causal and stochastic independence

1 Introduction

The aim of this work is to make the first steps towards a general theory of concurrent Markov Processes in *partially ordered logical time.* The original motivation for this approach is the diagnosis of distributed systems (see [8, 9]): In such systems, as they are found in telecommunications networks, faults will trigger alarms as well as other faults elsewhere in the system. The supervisors will then observe streams of alarm events coming in from different locations, with no global temporal ordering: only alarms from the same sensor will be totally ordered with respect to the local "logical clock" of that sensor. Events in different locations are ordered only if there exists a known causal chain between them. Since only the *alarms* are observable, these causalities are not directly observable but are part of the model used for diagnosis. The diagnosis task is to deduce the likely causes of the alarms, that is, the *faults*: given a partial order pattern of alarms, construct the *possible scenarios*, i.e. the partial order patterns of faults *(a)* that respect the known causalities in the system, and *(b)* that explain the observations, i.e. whose actual occurrence would have triggered the pattern of alarms received; then, find the *most likely* among these partial order scenarios of faults. The measure for this likelihood has to be given by an a priori probability measure on the different possible *partially ordered runs.*

The crucial point is that neither the *global time* nor the *global state* of the system are available; the temporal stochastic processes treated in the literature generally require both. This is the case also for models designed to reflect concurrency and non-determinism, such as *(Generalized) Stochastic Petri Nets* (e.g. [1]), *Stochastic Transition Systems* (de Alfaro[5, 6]) and *probabilistic automata* (Rabin [33]; Segala [35, 36, 37]), etc. *Markov decision processes*, see [13], arise from the asynchronous parallel composition of (discrete time) Markov chains, yielding again a process with global state and time. The local transition probabilities are chosen non-deterministically, according to some *policy* that may be randomized in its turn. The model we develop here, is a *distributed one;* however, one encounters the above stratification as well: in fact, there is an upper layer – the cluster net – controls the unfolding algorithm on the original net $\mathcal{N}$ by selecting the partial order of probabilistic choices, and a lower layer, that of $\mathcal{N}$ itself, where these choices are made, according to distributions given by the current local states.

Figure 1 shows a common Petri net situation, whose probabilistic treatment causes problems[1]. A natural first approach to randomizing the possible behaviors on the left hand side is to assign probabilities $\mathbf{p}_{\mathsf{A}}, \mathbf{p}_{\mathsf{B}}, \mathbf{p}_{\mathsf{C}}, \mathbf{p}_{\mathsf{D}}$ to each transition; $\mathbf{p}_{\mathsf{t}}$ denotes the probability that t will be the next transition to be fired. Then we have to require that $\mathbf{p}_{\mathsf{A}} + \mathbf{p}_{\mathsf{B}} + \mathbf{p}_{\mathsf{C}} + \mathbf{p}_{\mathsf{D}} \leqslant 1$; assume without loss of generality that $\mathbf{p}_{\mathsf{A}} + \mathbf{p}_{\mathsf{B}} + \mathbf{p}_{\mathsf{C}} + \mathbf{p}_{\mathsf{D}} = 1$, and that all $\mathbf{p}'$s are positive. This approach yields a probability for *sequences* rather than for *concurrent runs*: consider the run with A and C, i.e. the equivalence class or *trace* formed by the two interleavings AC and CA. Since both choices are made independently of one another (in the terminology below, the two *clusters* are independent), one would hope that the order of A and C does not matter: however, the probability that "A and then C" is

$$\frac{\mathbf{p}_{\mathsf{A}} \cdot \mathbf{p}_{\mathsf{C}}}{\mathbf{p}_{\mathsf{C}} + \mathbf{p}_{\mathsf{D}}},$$

[1] the example is also discussed in [4].

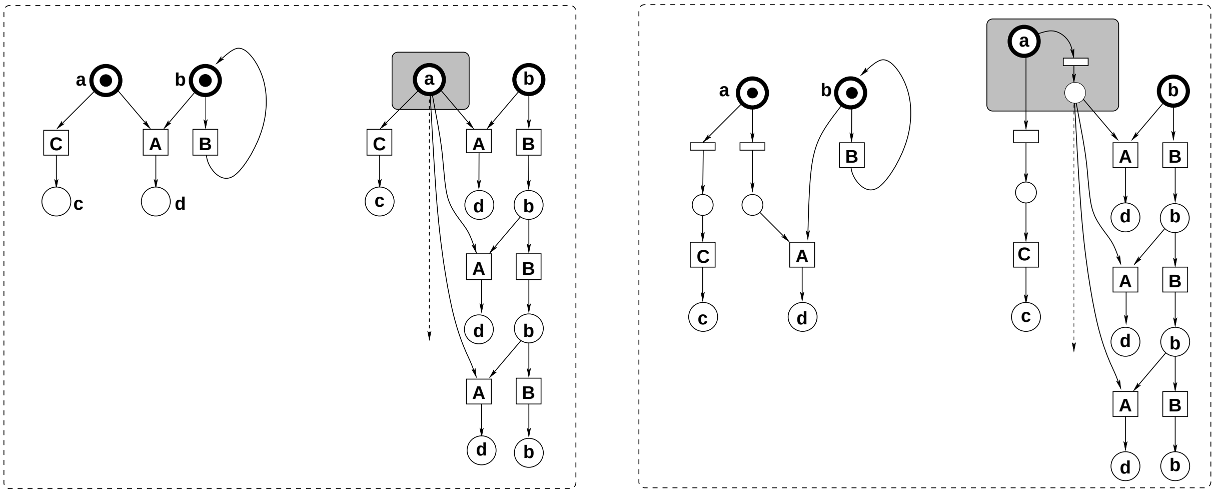

Figure 2: The limitations of the branching process semantics and the use of pre-selection (see [8])

whereas "C and then A" gives

$$\frac{\mathbf{p}_{\mathsf{A}} \cdot \mathbf{p}_{\mathsf{C}}}{\mathbf{p}_{\mathsf{A}} + \mathbf{p}_{\mathsf{B}}}.$$

The quotient of these two probabilities is unbounded and can take *any* positive value; this model thus neither respects, nor makes use of, the independence of the two components. When looking for a "truly local model", we have to beware, however, of non-local effects inevitable in Petri nets: on the right hand side of Figure 1, a place c is added, joining the two parts. Now, the dependence/independence of A and C depends on the marking of c: if c is empty, the discussion above remains valid; otherwise, there is an influence of A on the possibilities (and hence probabilities) for C, and vice versa: given the occurrence of A, the probability of C increases from $\mathbf{p}_C/[\mathbf{p}_C + \mathbf{p}_D]$ to $\mathbf{p}_C$. In the other direction, given the occurrence of C, the probability of A is even 1 ! (Note: This is a probabilistic view on what is called *confusion* in [38].

For modeling concurrent behavior, a natural framework is given by *unfolding* semantics using *occurrence Petri net.* In the literature, these are usually constructed using *branching processes* following [32]; see also [17, 20, 22, 30]. Examples for branching process are shown in the two parts of Figure 2: in both parts, the behavior of the Petri net on the left hand side is represented by an acyclic net on the right; we will comment on the mechanism of this construction below.

In [8], as well as [40], probabilistic unfoldings for 1-safe untimed Petri nets were introduced based on *routing measures.* This works with no difficulty in the context of Free Choice Nets, where all choices leading to branching behavior are local to the places. For general nets, however, the need for re-normalizing the routing probabilities for arbitrary net structures result in important restrictions to the approach of [8]. To see some of the difficulties, consider the left hand side of Figure 2. It shows a small Petri net $\mathcal{N}$ and a prefix of its branching process. Roughly speaking, every current or future token in the net is represented by a circle in the branching process net, and different outgoing arcs represent the different possibilities to consume that token. Now, branching of arcs in the unfolding may correspond to actual choices; this is the case for the arcs leaving b, or for the choice between A and C seen from a. However, there is no "choice" of a between the different A's, since all of those represent the consummation of the same token of a. So, assigning probabilities to the different branches of a creates technical as well as conceptual difficulties. (In fact, the right hand side of Figure 2 shows an attempt to solve these problems, see below). As a consequence, the approach remains limited to subclasses of Petri nets, outside of which it is not applicable.

The present article shows that the difficulties and restrictions can be overcome if the branching process semantics with its token-oriented view is replaced by a *cluster view.* Cluster processes were first proposed in [27]. The unfolding semantics obtained from it will here be used in constructing both the net unfolding and the probability measures. There is no structural restriction on the form of the net, nor any boundedness assumption; the approach applies to *any* Petri Net.

In studying the cluster semantics here, we will see below that cluster unfoldings need to be constructed under some controlling policy, which will be given by the *cluster net* introduced below. The semantics then leads in a natural way to a probabilistic model on the space of concurrent runs, with a variety of possibilities for randomizing locally inside the clusters; for this, we will present two examples of *Gibbs field* constructions that yield probabilities reflecting Petri net dynamics and locality. The final contributions here will then be the introduction and structural characterization of *logical stopping times*, and the proof of a strong Markov Property.

The paper is organized as follows. Section 2 reviews basic notations and definitions; in Section 3, we discuss and develop Petri net unfoldings under cluster semantics. The distributed nature of the unfolding algorithm leads us to introduce an auxiliary construction, the *cluster net*; it will schedule, from the top layer, the order of inspection (*policy*) of the clusters in the lower layer net, and thus the order in which those make their local choices. All policies for directing the unfolding process can be obtained as a "firing sequence" of that virtual net; moreover, it allows to structure the space of policies with a probability measure consistent with those used for the local choices. The probabilistic model is developed in Section 4; Section 5 introduces stopping times and proves the strong Markov Property, and Section 6 ends the paper with some comments and outlooks.

2 Petri Nets

2.1 Basic Definitions

We begin with the basic definitions that will be used throughout. $\mathbb{N}_0$ denotes the set of *non-negative* integers, and $\mathbb{N}$ that of the *positive* integers; $\mathbb{Z}$ is the set of (all) integers. For a set $\mathcal{X}$, denote the set of multi-sets $\mathsf{M} : \mathcal{X} \to \mathbb{N}_0$ over $\mathcal{X}$ as $\mathfrak{M}(\mathcal{X})$. Let $\mathsf{x} \in \mathcal{X}$. For $\mathsf{M} \in \mathfrak{M}(\mathcal{X})$, write $\mathsf{x} \in \mathsf{M}$ iff $\mathsf{M}(\mathsf{x}) > 0$; the *support* of M is the set $\mathbf{supp}(\mathsf{M}) \overset{\mathsf{def}}{=} \{\mathsf{x} \mid \mathsf{x} \in \mathsf{M}\}$.

If $\mathsf{K} \subseteq \mathcal{X} \times \mathcal{X}$ is a binary relation over $\mathcal{X}$, let $\mathsf{K}^{-1} \overset{\mathsf{def}}{=} \{(y, x) \mid x\mathsf{K}y\}$ and $\mathsf{K}[x] \overset{\mathsf{def}}{=} \{y \mid x\mathsf{K}y\}$; by extension, if $\mu \subseteq \mathcal{X}$, write $\mathsf{K}[\mu] \overset{\mathsf{def}}{=} \bigcup_{x \,\in\, \mu} \mathsf{K}[x]$. For a binary relation $\mathsf{K} \subseteq \mathcal{X} \times \mathcal{X}$, denote as K^* the reflexive transitive closure of K and as K^+ the irreflexive part $\mathsf{K}^+ \overset{\mathsf{def}}{=} \mathsf{K}^* \backslash \{(\mathsf{x}, \mathsf{x}) \mid \mathsf{x} \in \mathcal{X}\}$ of K^*. A *Petri net* (with *arc weights*) is a tuple of the form $\mathcal{N} = (\mathcal{P}, \mathcal{T}, \mathcal{W}, \mathsf{M})$. Here, $\mathcal{P} = \mathcal{P}(\mathcal{N})$ is a set of places and $\mathcal{T} = \mathcal{T}(\mathcal{N})$ a set of transitions such that $\mathcal{P} \cap \mathcal{T} = \emptyset$, and $\mathcal{P} \cup \mathcal{T}$ is finite[2]. As usual, the figures here show places as circles and transitions as rectangles. Further, $\mathcal{W} : ((\mathcal{P} \times \mathcal{T}) \cup (\mathcal{T} \times \mathcal{P})) \to \mathbb{N}_0$ is the *arc weight* function. The set F of *arcs* of $\mathcal{N}$ is given by $\mathsf{F} \overset{\mathsf{def}}{=} \mathcal{W}^{-1}(\mathbb{N})$; for a node $\mathsf{x} \in (\mathcal{P} \cup \mathcal{T})$, set ${}^\bullet\mathsf{x} \overset{\mathsf{def}}{=} \mathsf{F}^{-1}[\mathsf{x}]$, $\mathsf{x}^\bullet \overset{\mathsf{def}}{=} \mathsf{F}[\mathsf{x}]$, and ${}^\bullet\mathsf{x}^\bullet \overset{\mathsf{def}}{=} {}^\bullet\mathsf{x} \cup \{\mathsf{x}\} \cup \mathsf{x}^\bullet$. A net $\mathcal{N} = (\mathcal{P}, \mathcal{T}, \mathcal{W}, \mathsf{M}_0)$ such that $\mathcal{W}$ takes 0 and 1 as its only values, is called *ordinary*; we will note ordinary nets as $\mathcal{N} = (\mathcal{P}, \mathcal{T}, \mathsf{F}, \mathsf{M}_0)$. We write ${}^\odot\mathsf{p}$ and $\mathsf{p}^\odot$ for the multi-sets ${}^\odot\mathsf{p}, \mathsf{p}^\odot \in \mathfrak{M}(\mathcal{T})$ of input/output weights for p, i.e. given, for $\mathsf{t} \in \mathcal{T}$, by ${}^\odot\mathsf{p}(\mathsf{t}) \overset{\mathsf{def}}{=} \mathcal{W}(\mathsf{t}, \mathsf{p})$ and $\mathsf{p}^\odot(\mathsf{t}) \overset{\mathsf{def}}{=} \mathcal{W}(\mathsf{t}, \mathsf{p})$. A *marking* of N is a multi-set M of places; if $\mathsf{M}(\mathsf{p}) = k$, we say there are *k tokens* on place p; tokens are shown as black dots in the figures. $\mathsf{M}_0 \in \mathfrak{M}(\mathcal{P})$ is the *initial marking* of $\mathcal{N}$.

Transitions may *fire* one by one, or in multi-sets; any transition multi-set $\delta : \mathcal{T} \to \mathbb{N}_0$ is called a *step*. Denote as λ the *empty step*, i.e. $\lambda(\mathsf{t}) = 0$ for all $\mathsf{t} \in \mathcal{T}$. The set of steps of $\mathcal{N}$ is denoted $\mathfrak{S}(\mathcal{N})$. A step δ is *enabled* in a marking M, denoted $\mathsf{M} \xrightarrow{\delta}$, iff M has enough tokens on all $\mathsf{p} \in \mathcal{P}$ to satisfy the sum of demands from δ, regarded as a vector in $\mathbb{Z}^{\mathcal{T}}$, concerning p:

$$\mathsf{M}(\mathsf{p}) \quad \geqslant \quad \langle \mathsf{p}^\odot, \delta \rangle, \tag{1}$$

where $\langle \cdot, \cdot \rangle : \mathbb{Z}^{|\mathcal{T}|} \times \mathbb{Z}^{|\mathcal{T}|} \to \mathbb{Z}$ is the inner product of $\mathbb{Z}^{|\mathcal{T}|}$. Denote the set of steps enabled in a marking M as $\mathsf{Enabled}(\mathsf{M}) \overset{\mathsf{def}}{=} \{\delta \mid \mathsf{M} \xrightarrow{\delta}\}$. Of course, $\lambda \in \mathsf{Enabled}(\mathsf{M})$ for *any* marking M. Step δ *transforms* marking

[2] *unfoldings* for nets can be infinite net-like structures, see below.

M into marking M' , denoted $\mathsf{M} \xrightarrow{\delta} \mathsf{M}'$, iff *(i)* $\mathsf{M} \xrightarrow{\delta}$, and *(ii)* for all $\mathsf{p} \in \mathcal{P}$:

$$\mathsf{M}'(\mathsf{p}) \quad = \quad \left[\mathsf{M}(\mathsf{p}) - \langle \mathsf{p}^{\odot}, \delta\rangle\right] + \langle {}^{\odot}\mathsf{p}, \delta\rangle. \tag{2}$$

Again, we trivially have $\mathsf{M} \xrightarrow{\lambda} \mathsf{M}$ for *any* marking M. A marking M is *reachable* from M_0, denoted $\mathsf{M}_0 \xrightarrow{*} \mathsf{M}$, iff: (i) $\mathsf{M} = \mathsf{M}_0$, or (ii) there exists a *firing sequence* $\mathsf{M}_0 \xrightarrow{\delta_1} \mathsf{M}_1 \xrightarrow{\delta_2} \dots \xrightarrow{\delta_n} \mathsf{M}_{n+1} = \mathsf{M}$.

Remark 1. *$\mathcal{W}(x,y)$ indicates the number of tokens that pass through the arc (x,y) at every* firing*; if y is a transition, $\mathcal{W}(x,y)$ gives the number of tokens that need to be present on x to allow a firing of y, and that will be removed from x by that firing. By contrast, is y is a place and thus x a transition, there is no requirement on y for a firing of x; if x fires, it only depends on its* input *places (if any) for doing so, and puts the indicated number of new tokens onto y, which cannot refuse them. This asymmetry, reflecting the direction of time in the Petri net dynamics, leads us to consider arcs in $\mathcal{T} \times \mathcal{P}$ as* passive, *as opposed to the* active *ones in $\mathcal{P} \times \mathcal{T}$; if* $\mathsf{t}\ \mathsf{F}\ \mathsf{p}$ *but* $\neg(\mathsf{p}\ \mathsf{F}\ \mathsf{t})$*, we say that* p *is* passive *w.r.t. transition* t. *This distinction will play an important role in the unfolding semantics below.*

2.2 Occurrence nets, Prefixes, Configurations, and Runs

Some further preparations before defining *occurrence nets*, the semantic domain for unfoldings. A tuple of three sets $\overline{\mathsf{N}} = (\mathsf{B}, \mathsf{E}, \overline{\mathsf{F}})$ is called a *net* iff $\mathsf{B} \cap \mathsf{E} = \emptyset$, and $\overline{\mathsf{F}} \subseteq [(\mathsf{B} \times \mathsf{E}) \cup (\mathsf{E} \times \mathsf{B})]$. In particular, forgetting arc weights and marking of a Petri net yields a net; but nets are more general than that since finiteness is not required. $\overline{\mathsf{N}}' = (\mathsf{B}', \mathsf{E}', \overline{\mathsf{F}}')$ is a subnet of $\overline{\mathsf{N}}$ iff **(i)** $\mathsf{B}' \subseteq \mathsf{B}$, **(ii)** $\mathsf{E}' \subseteq \mathsf{E}$, and **(iii)** $\overline{\mathsf{F}}' = \overline{\mathsf{F}} \cap [(\mathsf{B}' \times \mathsf{E}') \cup (\mathsf{E}' \times \mathsf{B}')]$; for $\mathcal{A} \subseteq (\mathsf{B} \cup \mathsf{E})$, the subnet of $\overline{\mathsf{N}}$ spanned by $\mathcal{A}$ is denoted $\overline{\mathsf{N}}[\mathcal{A}]$. Write ${}^{\circ}\mathsf{x} \overset{\mathsf{def}}{=} \overline{\mathsf{F}}^{-1}[\mathsf{x}]$ and $\mathsf{x}^{\circ} \overset{\mathsf{def}}{=} \overline{\mathsf{F}}[\mathsf{x}]$; set $< \overset{\mathsf{def}}{=} \overline{\mathsf{F}}^{+}$ and $\leqslant = \overline{\mathsf{F}}^{*}$. The *conflict* relations ic and $\#$ are given by:

1. For $\mathsf{e}_1, \mathsf{e}_2 \in \mathsf{E}$, let $\mathsf{e}_1 \ \mathsf{ic}\ \mathsf{e}_2$ iff $\mathsf{e}_1 \neq \mathsf{e}_2$ and ${}^{\circ}\mathsf{e}_1 \cap {}^{\circ}\mathsf{t}_2 \neq \emptyset$.

2. For $x, y \in (\mathsf{B} \cup \mathsf{E})$, let $x \ \#\ y$ iff there exist $\mathsf{e}_1, \mathsf{e}_2 \in \mathsf{E}$: *(i)* $\mathsf{t}_1 \ \mathsf{ic}\ \mathsf{t}_2$, *(ii)* $\mathsf{t}_1 \leqslant x$, and *(iii)* $\mathsf{t}_2 \leqslant y$.

For $\mathcal{U} \subseteq (\mathsf{B} \cup \mathsf{E})$, write $\max_{<}(\mathcal{U})$ and $\min_{<}(\mathcal{U})$ for the set of maximal/minimal elements of $\mathcal{U}$ w.r.t. $<$, respectively. Now, we are ready to define:

Definition 1. *A net* $\overline{\mathsf{N}} = (\mathsf{B}, \mathsf{E}, \overline{\mathsf{F}})$ *is called a* pre-occurrence net *iff it satisfies:*

1. no backward branching: $|{}^{\circ}\mathsf{b}| \leqslant 1$ *for all* $\mathsf{b} \in \mathsf{B}$*;*

2. Acyclicity*: With "$<$" obtained from* $\overline{\mathsf{F}}$ *as above,* $\neg(x < x)$ *for all* $x \in (\mathsf{B} \cup \mathsf{E})$*;*

3. Absence of auto-conflict*:* $\neg(x \# x)$ *for all* $x \in (\mathsf{B} \cup \mathsf{E})$*;*

4. $\overline{\mathsf{N}}$ *is* condition-initialized*, i.e.* $\mathbf{c}_0 \overset{\mathsf{def}}{=} \min_{<}(\mathsf{B} \cup \mathsf{E}) \subseteq \mathsf{B}$.

The elements of b *are called* conditions*, those of* e *the* events *of* $\overline{\mathsf{N}}$. *An* occurrence net *or* ON *is a pre-occurrence net that, in addition, is*

1. well-founded*, i.e. there exists no infinitely $<$-decreasing sequence,*

2. condition-bordered*:* $\max_{<}(\mathsf{B} \cup \mathsf{E}) \subseteq \mathsf{B}$.

A causal net *or* CN *is a pre-occurrence net such that* $|\mathsf{b}^{\circ}| \leqslant 1$ *for all* $\mathsf{b} \in \mathsf{B}$.

The place-bordered requirement is non-standard but means no loss of generality; any net meeting all other requirements can be extended into a place-bordered one, without changing its other properties.

The following relations are useful in analyzing occurrence nets. The causal dependence relation is $\mathsf{li} \overset{\mathsf{def}}{=} < \cup <^{-1}$; with the identity relation $\mathsf{id} \overset{\mathsf{def}}{=} \{(x,x)\ :\ x \in \mathsf{B} \cup \mathsf{E}\}$, define elementary concurrency as $\mathsf{co} \overset{\mathsf{def}}{=} (\mathsf{B} \cup \mathsf{E})^2 - (\mathsf{li} \cup \# \cup \mathsf{id})$. One obtains easily that **(i)** $<$ is a partial order; **(ii)** li, co, ic and $\#$ are symmetric and irreflexive; **(iii)** a pre-occurrence net $\overline{\mathsf{N}}$ is a CN iff $\# = \emptyset$.For a binary relation K, denote as *kens* of K its maximal cliques w.r.t. set inclusion, and the set of K-kens as $\mathcal{K}(\mathsf{K})$. A co−clique $\mathcal{X} \subseteq \mathsf{B}$ 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 $\mathfrak{C}(\overline{\mathsf{N}})$ are called *cuts*. The cuts consisting only of conditions, i.e. maximal co-sets, will be called *condition-cuts*; the set of condition-cuts is denoted $\mathfrak{C}_{\mathcal{P}}(\overline{\mathsf{N}})$. The condition-cuts are exactly the maximal co-sets. In particular, $\mathbf{c}_0 = \min_<(\mathsf{E} \cup \mathsf{B}) = \min_<(\mathsf{B})$ is a condition-cut.

Now, let $\overline{\mathsf{N}}$ be an occurrence net. For a node x, denote as $\mathsf{x}^{\downarrow} \overset{\mathsf{def}}{=} \{y \mid y < \mathsf{x}\}$ and $\mathsf{x}^{\Downarrow} \overset{\mathsf{def}}{=} \{y \mid y \leqslant \mathsf{x}\}$ the *strict past* and *past* of x, and similarly, the *strict future* and *future* of x as $\mathsf{x}^{\uparrow} \overset{\mathsf{def}}{=} \{y \mid \mathsf{x} \ < \ y\}$ and $\mathsf{x}^{\Uparrow} \overset{\mathsf{def}}{=} \{y \mid \mathsf{x} \ \leqslant \ y\}$. Further, denote the *neighborhood* of node x as ${}^\circ\mathsf{x}^\circ \overset{\mathsf{def}}{=} {}^\circ\mathsf{x} \cup \{\mathsf{x}\} \cup \mathsf{x}^\circ$. For any set $\mathcal{X}$ of nodes of $\overline{\mathsf{N}}$, call $\mathcal{OH}(\mathcal{X}) \overset{\mathsf{def}}{=} \mathcal{X} \ \cup \ 9\bigcup^{\circ}_{\mathsf{x} \in (\mathcal{X} \cap \mathsf{e})} \mathsf{x}^\circ)$ the *place-bordered* or *open hull* of $\mathcal{X}$. To compare *sets* rather than individual nodes, we shall use the *Egli-Milner order* $\sqsubseteq$: for $\mathcal{X}_1, \mathcal{X}_2 \subseteq (\mathsf{B} \cup \mathsf{E})$,

$$\mathcal{X}_1 \ \sqsubseteq \ \mathcal{X}_2 \quad \overset{\mathsf{def}}{\Longleftrightarrow} \quad [(\mathsf{x}_1 \in \mathcal{X}_1 \wedge \mathsf{x}_2 \in \mathcal{X}_2) \Rightarrow \mathsf{x}_2 \ \not< \ \mathsf{x}_1] \wedge \ [\forall\, \mathsf{x}_2 \in \mathcal{X}_2 \ \exists\, \mathsf{x}_1 \in \mathcal{X}_1 : \ \mathsf{x}_1 \ \leqslant \ \mathsf{x}_2]. \qquad (3)$$

Definition 2. *A* prefix *of* $\overline{\mathsf{N}} = (\mathsf{B}, \mathsf{E}, \overline{\mathsf{F}})$ *is any set*[3] $\mathcal{U} \subseteq (\mathsf{B} \cup \mathsf{E})$ *that is:*

1. causally closed*: If* $x \in \mathcal{U}$*, then* $x^{\downarrow} \subseteq \mathcal{U}$*;*

2. condition-bordered*:* $\mathcal{U} = \mathcal{OH}(\mathcal{U})$*; and*

3. condition-initialized*:* $\mathbf{c}_0 \subseteq \mathcal{U}$*.*

The set of prefixes of $\overline{\mathsf{N}}$ *is denoted* Pref*. A* configuration *is a conflict-free prefix* $\mathbf{C}$*, i.e. such that* $\# \cap (\mathbf{C} \times \mathbf{C}) = \emptyset$*. Denote the set of configurations of* $\overline{\mathsf{N}}$ *as* $\mathbb{C}(\overline{\mathsf{N}})$*. Let* $\Omega(\mathcal{N})$ *be the set of maximal cliques of* $\mathsf{tg} \overset{\mathsf{def}}{=} (\mathsf{co} \ \cup \ \mathsf{li})$*; the elements of* $\Omega(\mathcal{N})$ *are called the* runs *of* $\overline{\mathsf{N}}$*.*

Every configuration is a causal net. One has $\Omega \subseteq \mathbb{C}$, and the runs are exactly the maximal configurations ([26]). For any configuration $\mathbf{C}$ and node x, $\mathcal{OH}(\mathbf{C} \ \cap \ \mathsf{x}^{\downarrow})$ and $\mathcal{OH}(\mathbf{C} \ - \ \mathsf{x}^{\Downarrow})$ are configurations; so is the intersection of a configuration with a prefix, which includes the case of the intersection of two configurations. Unions *and* intersections of prefixes yield again prefixes; in fact, $(\mathsf{Pref}, \cup, \cap)$ is a *complete* lattice. Unions of configurations, however, do not yield configurations in general.

2.3 Unfoldings

Unfoldings reflects both concurrent and branching behavior of a general marked Petri net in the structure of an occurrence net. There are different rules for these *unfoldings* (cf. [17, 25, 26, 27, 32, 41]). Every unfolding is an occurrence net generated inductively by a net system, reflecting the initial marking by the initial cut and representing subsequent firings of transitions by *event* nodes, and of subsequent place markings by *conditions*.

2.3.1 Branching Processes

First, recall the most widely used occurrence net semantics, *branching processes*, shown as *(I)* in Figurefig:views ; the following definition is slightly more general than those given in [11, 17, 22, 32] since it allows for arbitrary initial markings.

Definition 3. *Let* $\mathcal{N} = (\mathcal{P}, \mathcal{T}, \mathsf{F}, \mathsf{M}_0)$ *be an ordinary Petri net and* $\overline{\mathsf{N}} = (\mathsf{B}, \mathsf{E}, \overline{\mathsf{F}})$ *an occurrence net.* $\mathcal{U} = (\overline{\mathsf{N}}, \mathbf{c}_0, \pi)$ *is a* branching process *of* $\mathcal{N}$ *iff* $\pi : (\mathsf{B} \cup \mathsf{E}) \to (\mathcal{P} \cup \mathcal{T})$ *satisfies:*

(i) $\pi(\mathsf{B}) \subseteq \mathcal{P}$ *and* $\pi(\mathsf{E}) \subseteq \mathcal{T}$*;*

(ii) for any $\mathsf{e} \in \mathsf{E}$*,* π *induces a graph isomorphism* π_{e} *between* $\overline{\mathsf{N}}_{\mathsf{e}} \overset{\mathsf{def}}{=} \overline{\mathsf{N}}[{}^\circ\mathsf{e}^\circ]$ *and* $\mathcal{N}_{\mathsf{e}} \overset{\mathsf{def}}{=} \mathcal{N}[{}^\bullet\pi(\mathsf{e})^\bullet]$*;*

(iii) for all $\mathsf{p} \in \mathcal{P}$*,* $\mathbf{c}_0$ *represents the number of tokens on* p *under* M_0*, i.e.* $|\pi^{-1}(\mathsf{p}) \cap \mathbf{c}_0| = \mathsf{M}_0(\mathsf{p})$*; and*

(iv) Irreducibility*:* $\forall \mathsf{e}_1, \mathsf{e}_2 \in \mathsf{E}$*,* ${}^\circ\mathsf{e}_1 = {}^\circ\mathsf{e}_2$ *and* $\pi(\mathsf{e}_1) = \pi(\mathsf{e}_2)$ *together imply that* $\mathsf{e}_1 = \mathsf{e}_2$*.*

[3] By abuse of terminology, we will also call $\overline{\mathsf{N}}[\mathcal{U}]$ a prefix

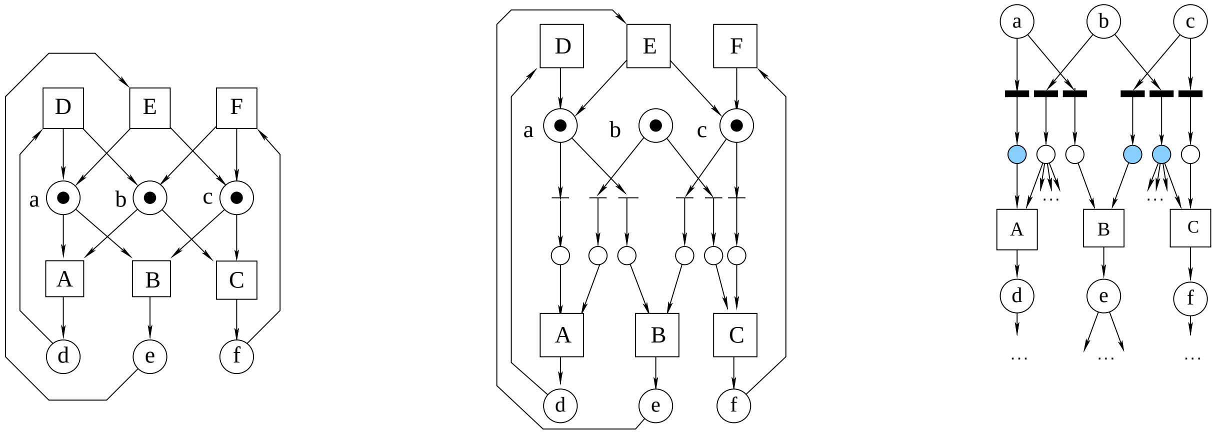

Figure 3: A net (left) for which pre-selection falsifies the behavior (center, right; see [8])

Thus every *condition* $\mathsf{b} \in \mathsf{B}$ in $\overline{\mathsf{N}}$ represents a token on the place $\mathsf{p} = \pi(\mathsf{b})$, and different arcs leaving a condition represent the different possibilities to consume that token. These different *events* can be firing instances of different transitions, or different firing instances, *events* of the same transition, see (I) in Figure 5. In [8], the probabilistic model is based on this semantics, and uses *routing measures*. These measures assign, for a place p, probabilities to all transitions $\mathsf{t} \in \mathsf{p}^\bullet$, indicating the probability that a token in p is *routed* toward t; the successive choices and firings yield a configuration of the system. The probability should then be given by the product of that of the individual choices. While this is natural and simple if all probabilistic decisions are made in *Free Choice* conflicts – this is the case of [40] where all other conflicts are non-randomized –, adjustments have to be made in more general nets: one has to restrict the probability onto the *coherent* realizations, i.e. such that the choices of different routers agree. To see the point, consider the left hand side of Fig. 4. Suppose a chooses B, b chooses D, and c selects C (one can assume this has positive probability). Then the net is blocked, despite a wealth of firing possibilities. Obviously, the probability mass lost due to such "runs" must be compensated by renormalization.

This is solved in [8] by identifying substructures – *layers* – of the unfolding, on which the corrected probability is determined; these layers then serve as building blocks in extending the probability to the full space of system runs. There are several limitations to this approach (see [8]). For an illustration, consider Figure 2: the branching of place a are not respected by the unfolding, new branches are added that do not correspond to actual choices. We refer to this phenomenon as *arc-scattering*: two outgoing arcs of place a, corresponding to options C and A in the *net*, become an infinity of arcs in the *unfolding*, without giving actual new choices to a. In the unfolding, an infinite number of instances of A all use the same condition, i.e. the same token on a, and are distinguished only by the number of times b chooses B before selecting A. So the probabilistic choice of a is diffused over an infinite set of events, among which a can not choose itself (we say that the unfolding is not *choice-conformal*), so the renormalization fails in practice. The *pre-selection* method indicated on the right hand side of Figure 2 helps in this case: by introducing *dummy* places that are not probabilized, one preserves the probabilistic choices of a without introducing new behaviors. Note that there is, in fact, no need for a probabilization of the dummy place since all choices are already determined by the router of b.

But if pre-selection helps avoid arc-scattering in some cases, it creates new difficulties in others. Consider the net in Figure 3. Here, the three dummy places necessary to ensure choice-conformalness give raise to a new possible system run which is blocked in a marking involving dummy places (indicated in black), while the original system is live. There are still more difficulties, discussed in [8]; in our view, these are intrinsic to the branching process semantics. Therefore, we shall be using an alternative semantics, better adapted to probabilization.

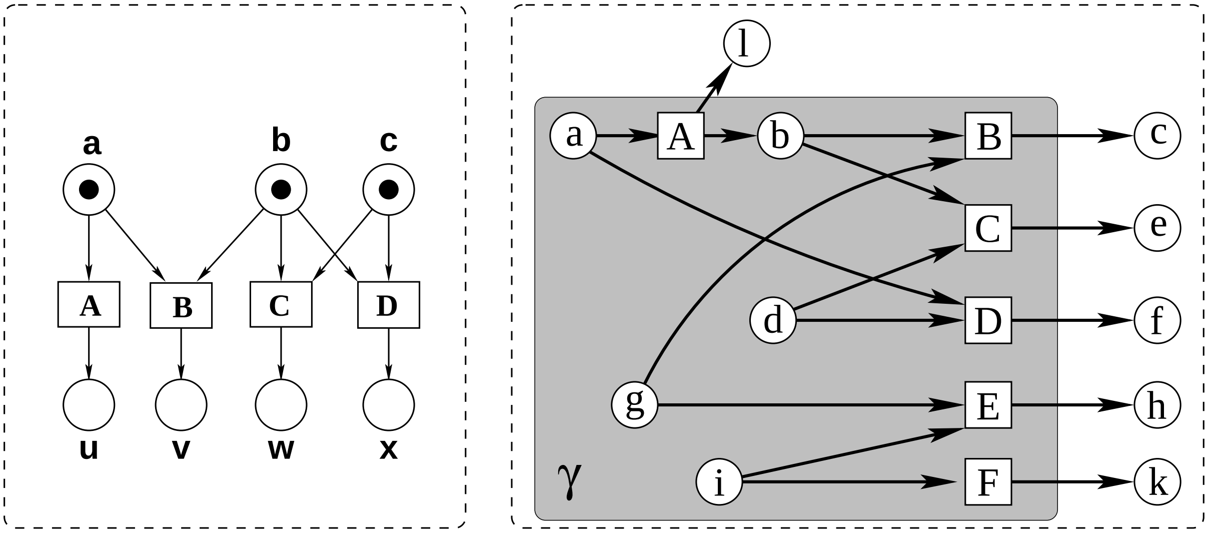

Figure 4: Left: On coordination of routing; right: a cluster

2.3.2 Changing the View: Cluster Semantics

In fact, there is not *one* but a multitude of possible unfolding semantics; the choice of a semantics is the choice of a *view* on PN behavior. Recall that *branching processes* are driven, informally speaking, by "token trajectories" and permit concurrency of events that do not compete for any individual token; they reflect the *individual token view.* The *collective token view* leads to *branching execution* semantics (Vogler [41], Esparza, Römer and Vogler [21], Haar [25]). It regards places as variables whose values are given by the number of tokens; transitions then read from and write on these variables in mutually exclusive access. As a result, *auto-concurrency* is excluded, i.e. no transition can fire more than once at a time, even if the marking would allow several concurrent firings; also, transitions accessing the same place may not fire jointly. This semantics does not reflect well enough the firing behavior to be a basis for a probabilistic model, hence we will not be using that semantics, either.

In *cluster* semantics, shown in (II) of Figure 5, we use the fact that $\mathcal{N}$ is naturally partitioned into node sets that are "minimally closed under conflicts", to construct unfoldings composed by local choices of steps. These sets are the *clusters* of $\mathcal{N}$ according to the following definition (see [14]):

Definition 4. *The* cluster $\gamma(\mathsf{x})$ *of* $\mathsf{x} \in (\mathcal{P} \cup \mathcal{T})$ *is the smallest set containing* x *that satisfies:*

$$\forall\ \mathsf{t} \in \mathcal{T}: \quad {}^{\bullet}\mathsf{t} \cap \gamma(\mathsf{x}) \neq \emptyset \quad \Rightarrow \quad \mathsf{t} \in \gamma(\mathsf{x}); \tag{4}$$

$$\forall\ \mathsf{p} \in \mathcal{P}: \quad \mathsf{p}^{\bullet} \cap \gamma(\mathsf{x}) \neq \emptyset \quad \Rightarrow \quad \mathsf{p} \in \gamma(\mathsf{x}) \tag{5}$$

By extension, call cluster *any set* $\gamma \subseteq (\mathcal{P} \cup \mathcal{T})$ *for which there exist* $x \in (\mathcal{P} \cup \mathcal{T})$ *such that* $\gamma = \gamma(x)$*; denote the set of clusters of* $\mathcal{N}$ *as* $\Gamma(\mathcal{N})$.

See the right hand side of Figure 4 for an illustration. We will unfold $\mathcal{N}$ in such a way that the events of the unfolding represent instances not necessarily of *single* transitions but of *steps.* So, *joint* firing of concurrent transition instances will be reflected, in all of the above cases, by a single event. However, using *global* steps would be unwise from a computational point of view, and also ignore the actual independence of events at a great distance (in terms of causal influence) from one another. Hence, rather than looking at *all* steps enabled in some global state of the net, examine the *local steps* within each *cluster*; once enabled, they can be fired irrespective of the behavior of other clusters, and allow to calculate the global steps, if desired, as their multi-set sums.

Definition 5. *For* $\gamma \in \Gamma(\mathcal{N})$*, write* $\mathcal{P}_\gamma \overset{\mathsf{def}}{=} \mathcal{P} \cap \gamma$ *and* $\mathcal{T}_\gamma \overset{\mathsf{def}}{=} \mathcal{T} \cap \gamma$. *A* γ-step *is a multi-set over* $\mathcal{T}_\gamma$*; let* $\mathfrak{S}(\gamma)$ *be the set of* γ*-steps, and* $\mathfrak{S}_\Gamma(\mathcal{N}) \overset{\mathsf{def}}{=} \bigcup\{\mathfrak{S}(\gamma) \mid \gamma \in \Gamma(\mathcal{N})\}$.

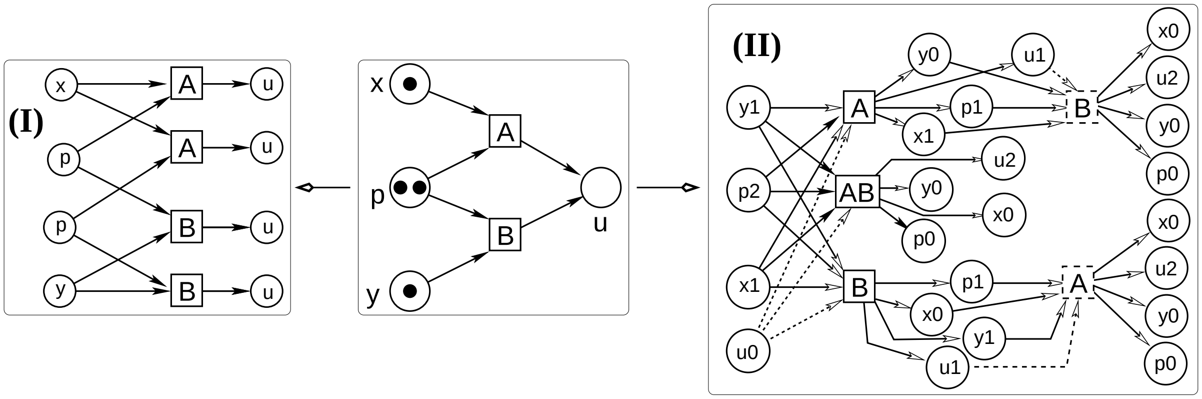

Figure 5: Engelfriet branching Process (I) and Cluster Process (II)

Thus, $\mathfrak{S}_\Gamma(\mathcal{N}) \subseteq \mathfrak{S}(\mathcal{N})$, and the inclusion is proper in general. We will denote as

$$\mathbf{in}(\gamma) \overset{\mathsf{def}}{=} \mathcal{P}_\gamma = {}^\bullet\gamma \quad \textit{and} \quad \mathbf{out}(\gamma) \overset{\mathsf{def}}{=} \{p \in (\mathcal{P}\backslash\mathcal{P}_\gamma) \mid {}^\bullet p \cap \gamma \neq \emptyset\}$$

the sets of *input* and *output* places of a cluster γ, respectively. Note that $\mathbf{out}(\gamma)$ is in general a proper subset of $\gamma^\bullet$: in the cluster on the right hand side of Figure 4, place b belongs to $\gamma^\bullet$ but not to $\mathbf{out}(\gamma)$. Clusters interact with one another by places; hence, call two clusters γ_1 and γ_2 *independent*, written $\gamma_1 \mathbb{I} \gamma_2$, iff the set of **s**hared **p**laces is empty:

$$\mathsf{sp}\,(\gamma_1, \gamma_2) \overset{\mathsf{def}}{=} (\mathbf{out}(\gamma_1) \cup \mathcal{P}_{\gamma_1}) \cap (\mathbf{out}(\gamma_2) \cup \mathcal{P}_{\gamma_2}) \quad = \quad \emptyset;$$

otherwise they are *dependent*, written $\gamma_1 \mathbb{D} \gamma_2$. Note that, by the definition of clusters, $\gamma_1 \neq \gamma_2$ implies that for any $\mathsf{x} \in \mathsf{sp}(\gamma_1, \gamma_2)$, x is a passive place for at least one of the two clusters γ_1 and γ_2.

Definition 6. (Cluster processes) *Let* $\mathcal{N} = (\mathcal{P}, \mathcal{T}, \mathcal{W}, \mathsf{M}_0)$ *be a Petri net,* $\overline{\mathsf{N}} = (\mathsf{B}, \mathsf{E}, \overline{\mathsf{F}}, \mathbf{c}_0)$ *an occurrence net, and let* $\pi : \mathsf{B} \to \mathcal{P}$, $\mu : \mathsf{B} \to \mathbb{N}_0$ *and* $\beta : \mathsf{E} \to \mathfrak{S}_\Gamma(\mathcal{N})$ *be mappings. Then* $\Pi \overset{\mathsf{def}}{=} (\overline{\mathsf{N}}, \pi, \beta, \mu)$ *is a* **cluster process** *of* $\mathcal{N}$ *iff*

1. Irreducibility: *for all* $\mathsf{e}_1, \mathsf{e}_2 \in \mathsf{E}$, $({}^\circ\mathsf{e}_1 = {}^\circ\mathsf{e}_2)$ *and* $(\beta(\mathsf{e}_1) = \beta(\mathsf{e}_2))$ *together imply* $\mathsf{e}_1 = \mathsf{e}_2$*;*
2. the initial cut reflects the initial marking*: for all* $\mathsf{b} \in \mathbf{c}_0$*,* $\mu(\mathsf{b}) = \mathsf{M}_0(\pi(\mathsf{b}))$*;*
3. π is injective on co-sets: *for all* $\mathsf{b}_1, \mathsf{b}_2 \in \mathsf{B}$ *such that* b_1 co b_2*, one has* $\pi(\mathsf{b}_1) \neq \pi(\mathsf{b}_2)$*;*
4. Events represent steps*: for all* $\mathsf{e} \in \mathsf{E}$*,*
   (a) ${}^\bullet\beta(\mathsf{e}) = \pi({}^\circ\mathsf{e})$ *(this condition is missing in [27]), and*
   (b) *for any* $\mathsf{p} \in \mathcal{P}$ *such that* ${}^\bullet\mathsf{p}^\bullet$ *contains a* $\mathsf{t_p} \in \delta \overset{\mathsf{def}}{=} \beta(\mathsf{e})$*, there exist* $\mathsf{b}_{in}, \mathsf{b}_{out} \in \mathsf{B}$ *such that:*
      i. ${}^\bullet\mathbf{supp}(\delta) \ \cap \ \pi^{-1}(\{\mathsf{b}\}) \ = \ \{\mathsf{b}_{in}\}$ *and* $\mathbf{supp}(\delta)^\bullet \ \cap \ \pi^{-1}(\{\mathsf{b}\}) \ = \ \{\mathsf{b}_{out}\}$*,*
      ii. $\mu(\mathsf{b}_{in}) \geqslant \langle \pi(\mathsf{b})^{\odot}, \delta \rangle$*, and*
      iii. $\mu(\mathsf{b}_{out}) = (\mu(\mathsf{b}_{in}) - \langle \pi(\mathsf{b})^{\odot}, \delta \rangle) + \langle {}^{\odot}\mathsf{b}, \delta \rangle$*.*

If $\Pi = (\overline{\mathsf{N}}, \pi, \beta, \mu)$ *and* $\Pi' = (\overline{\mathsf{N}}', \pi', \beta', \mu')$ *are two processes of* $\mathcal{N}$ *such that* $\overline{\mathsf{N}}'$ *is a subnet of* $\overline{\mathsf{N}}$*, and* π', β', μ' *are the restrictions of* π, β, μ *to* $\overline{\mathsf{N}}'$*, respectively, we call* Π' *a* prefix *of* Π.

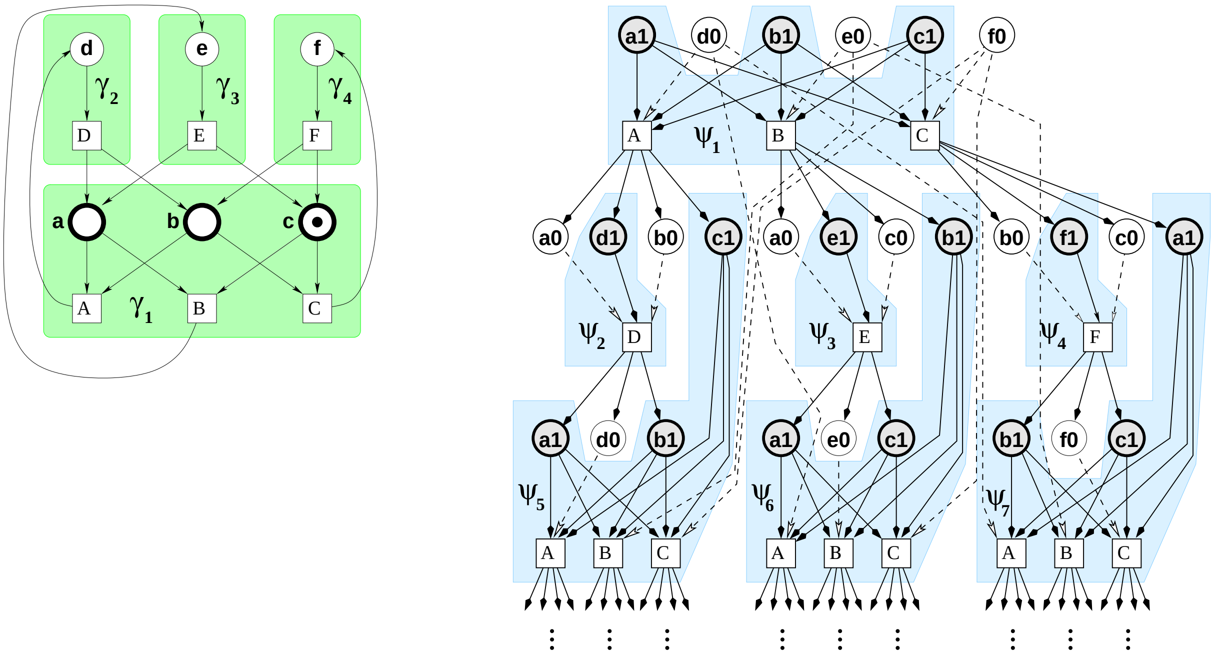

Figure 6: A Cluster process (λ-events omitted) for Fig. 3. Shaded areas show clusters (left) and tiles (right).

The mappings π and β correspond to the structural unfolding, taking conditions to places and events to cluster steps; since conditions represent *states* of places, we also need the mapping μ to assign token numbers to conditions. For Part 4b of Definition 6, compare the firability condition (1) and the firing equation (2). We claim that to each condition-cut $\mathbf{c}$ of the cluster unfolding corresponds a unique reachable marking $\mathsf{M}(\mathbf{c})$ of $\mathcal{N}$, such that for two cuts $\mathbf{c}_1, \mathbf{c}_2$ and an event e, one has

$$\mathbf{c}_2 = (\mathbf{c}_1 \backslash {}^\circ\mathsf{e}) \cup \mathsf{e}^\circ \quad \Rightarrow \quad \mathsf{M}(\mathbf{c}_1) \xrightarrow{\beta(\mathsf{e})} \mathsf{M}(\mathbf{c}_2). \tag{6}$$

If the left hand side of (6) is satisfied, write $\mathbf{c}_1 \overset{\mathsf{e}}{\rightsquigarrow} \mathbf{c}_2$; write $\mathbf{c} \rightsquigarrow \mathbf{c}'$ iff there are cuts $\mathbf{c} = \mathbf{c}_1, \ldots, \mathbf{c}_n = \mathbf{c}'$ and events $\mathsf{e}_1, \ldots, \mathsf{e}_{n-1}$ such that $\mathbf{c}_i \overset{\mathsf{e}_i}{\rightsquigarrow} \mathbf{c}_{i+1}$ for $i \in \{1, \ldots, n-1\}$. The following property is characteristic of cluster semantics; no equivalent holds for branching processes.

Lemma 1. *If all cuts are finite, every condition-cut* $\mathbf{c}$ *is in bijection via* π *with* $\mathcal{P}$*.*

> **Proof:** Fix $\mathbf{c}$; it suffices to show $\mathcal{P} \subseteq \pi(\mathbf{c})$. By definition, this holds for $\mathbf{c} = \mathbf{c}_0$. Since all cuts are finite, $\mathbf{c}_0 \rightsquigarrow \mathbf{c}$ (see [11]), and the claim follows by induction using Definition 6. □

Now, letting $\mathsf{b}_{\mathbf{c}}(\mathsf{p})$ be the unique condition in $\mathbf{c}$ such that $\pi(\mathsf{b}_{\mathbf{c}}) = \mathsf{p}$, and setting $\mathsf{M}(\mathbf{c})(\mathsf{p}) \overset{\mathsf{def}}{=} \mu(\mathsf{b}_{\mathbf{c}}(\mathsf{p}))$, we obtain in fact (6), using 4b of Definition 6. As a consequence, every condition-cut $\mathbf{c}$ and every finite configuration $\mathbf{C}$ correspond to a unique marking, $\mathsf{M}_{\mathbf{c}}$ and $\mathsf{M}_{\mathbf{C}} = \mathsf{M}_{\max_{\sqsubseteq}(\mathbf{C})}$, respectively; on the other hand, the same M may be obtained from different configurations.

2.4 Tiles

A process net $\overline{\mathsf{N}}$ is composed of basic pieces given by the clusters. In fact, let γ be some cluster and e be an event corresponding to a step $\delta \in \mathfrak{S}(\gamma)$; then there is only one cluster γ with this property, uniquely determined by e, so we denote it by $\gamma(\mathsf{e})$. Moreover, ${}^\circ\mathsf{e}$ contains a complete copy under π^{-1} of $\mathbf{in}(\gamma(\mathsf{e}))$. All events e' that share at least one pre-condition from $\pi^{-1}(\mathbf{in}(\gamma(\mathsf{e})))$ with e *and* that belong to the same cluster, i.e. $\gamma(\mathsf{e}') = \gamma(\mathsf{e})$, will share with e *all* the pre-conditions corresponding to places of $\mathbf{in}(\gamma(\mathsf{e}))$. On the other hand, the corresponding steps of $\mathcal{N}$ have, in general, different output places or values, so the corresponding events will differ in their presets by conditions that need not correspond to places in $\mathbf{in}(\gamma(\mathsf{e}))$. This motivates the following definition (see Figure 6):

Definition 7. *The* tile *of* e *and its* completion *are defined as, respectively,*

$$\begin{aligned}\psi(\mathsf{e}) &\overset{\mathsf{def}}{=} ({}^\circ\mathsf{e} \cap \pi^{-1}(\mathbf{in}(\gamma(\mathsf{e})))) \cup \{\mathsf{e}' \in \mathsf{E} \mid \gamma(\mathsf{e}') = \gamma(\mathsf{e}) \wedge \ \mathbf{in}(\gamma(\mathsf{e})) \subseteq \pi({}^\circ\mathsf{e} \cap {}^\circ\mathsf{e}')\} \\ \psi_*(\mathsf{e}) &\overset{\mathsf{def}}{=} \psi(\mathsf{e}) \cup {}^\circ\psi(\mathsf{e}).\end{aligned}$$

A subnet ψ (ψ_*) *of* $\overline{\mathsf{N}}$ *is called a* tile *(*complete tile*) iff there exists an event* $\mathsf{e} \in \mathsf{E}$ *such that* $\psi = \psi(\mathsf{e})$ $(\psi = \psi(\mathsf{e}))$*; the set of (complete) tiles of* $\mathcal{N}$ *is denoted* Ψ (Ψ_*) .

All events of a tile are pairwise in conflict:

Lemma 2. *For any* $\mathsf{e}, \mathsf{e}' \in \mathsf{E}$*, if (i)* $\mathsf{e} \neq \mathsf{e}'$ *and (ii)* $\exists \mathsf{e}'' \in \mathsf{E}:\ \mathsf{e}, \mathsf{e}' \in \psi(\mathsf{e}'')$*, then* $\mathsf{e} \ \mathsf{ic}\ \mathsf{e}'$ *and hence* $\mathsf{e} \ \#\ \mathsf{e}'$.

> **Proof:** Suppose there exists a tile ψ and events $\mathsf{e}, \mathsf{e}' \in \psi \cap \mathsf{E}$ such that ${}^\circ\mathsf{e} \cap {}^\circ\mathsf{e}' = \emptyset$, and let $\mathsf{e}'' \in \mathsf{E}$ such that $\mathsf{e}, \mathsf{e}' \in \psi(\mathsf{e}'')$. By the definition of the unfolding, $\pi^{-1}(\mathbf{in}(\gamma(\mathsf{e}))) \cap {}^\circ\mathsf{e} \cap {}^\circ\mathsf{e}'' \neq \emptyset$ implies that $\pi^{-1}(\mathbf{in}(\gamma(\mathsf{e}))) \cap {}^\circ\mathsf{e} = \pi^{-1}(\mathbf{in}(\gamma(\mathsf{e}))) \cap {}^\circ\mathsf{e}''$; by the same argument, $\pi^{-1}(\mathbf{in}(\gamma(\mathsf{e}))) \cap {}^\circ\mathsf{e}' = \pi^{-1}(\mathbf{in}(\gamma(\mathsf{e}))) \cap {}^\circ\mathsf{e}''$, from which the claim follows. □

Note that a tile is, in general, *not* a cluster of $\overline{\mathsf{N}}$: In Figure 6, condition d_0 does not belong to any of the tiles ψ_1, ψ_6 or ψ_7 containing post-events of d_0; this is due to the fact that d_0 reflects an *output* place of γ_1. In fact, $d_0{}^\circ$ is infinite; this does, however, not cause a problem for the probabilization since d_0's outgoing arcs are passive, see below. However, a tile ψ represents all possible actions of the cluster $\gamma(\psi)$ given by $\gamma(\mathsf{e})$ for any event e of ψ, under the local marking M obtained as

$$\forall \mathsf{p} \in \mathbf{in}(\gamma(\psi)):\ \mathsf{M}(\mathsf{p}) \ \overset{\mathsf{def}}{=}\ \mu(\mathsf{b}),$$

where b is the unique condition of ψ satisfying $\pi(\mathsf{b}) = \mathsf{p}$.

The strong flow relation. Now, a further look at local *causality* between places and transitions. Into the unfolding net, the cluster unfolding semantics puts arcs from condition b to event e if the corresponding place $\mathsf{p} \overset{\mathsf{def}}{=} \pi(\mathsf{b})$

(A) belongs to the same cluster γ as $\beta(\mathsf{e})$ and thus has an impact on its probability, or

(B) p is passive w.r.t. all transitions of γ, i.e. $\mathsf{p} \notin \gamma$, but *receives tokens* when δ is fired.

Since passive places have no causal influence on the choice of steps, we consider as *causal* only those arcs that do *not* belong to type 2.4 (B), and distinguish the corresponding arcs in the unfolding. Define the *strong flow relation* of $\overline{\mathsf{N}}$ as

$$\mathbb{F} \ \overset{\mathsf{def}}{=}\ \overline{\mathsf{F}} - \{(\mathsf{b}, \mathsf{e}) \in (\mathsf{B} \times \mathsf{E}) \mid \gamma\,(\pi(\mathsf{b})) \neq \gamma(\mathsf{e})\}\,;$$

in Figure 6, $\mathbb{F}$ is drawn in solid lines. The $\mathbb{F}$-arcs are those that correspond either to a transition output or to a *proper* transition *in*put, i.e. one that actually removes tokens. Since $\mathbb{F} \subseteq \overline{\mathsf{F}}$, one obtains a partial order $\prec$ by setting $\prec \overset{\mathsf{def}}{=} \mathbb{F}^+$; obviously, $\mathbb{F}^+ \subseteq \overline{\mathsf{F}}^+$. Observe that all tiles are *oblong* w.r.t. $\prec$: one has $\mathsf{b} \prec \mathsf{b}'$ for any $\mathsf{b} \in {}^\circ\psi$ and $\mathsf{b}' \in \mathbb{F}[\psi]$. We cannot extend this statement to "$<$" (adding dotted in-arcs to the tiles) since, as Figure 6 shows, the oblong property would otherwise be lost: Condition $f0$ precedes the C-event of ψ_5, but not the A in ψ_5. Together with Lemma 2, the fact that tiles are oblong immediately implies the following property, which will help characterize *stopping times* below:

Lemma 3. *For any cut* $\mathbf{c} \in \mathfrak{C}(\overline{\mathsf{N}})$ *and* $\psi \in \Psi(\overline{\mathsf{N}})$*, one has* $[(\mathbf{c} \cap \psi) \neq \emptyset] \Rightarrow [(\mathbf{c} \cap \mathbb{F}[\psi]) = \emptyset]$.

That is, $\mathbf{c}$ may contain either entry or exit conditions of a tile, but not both.

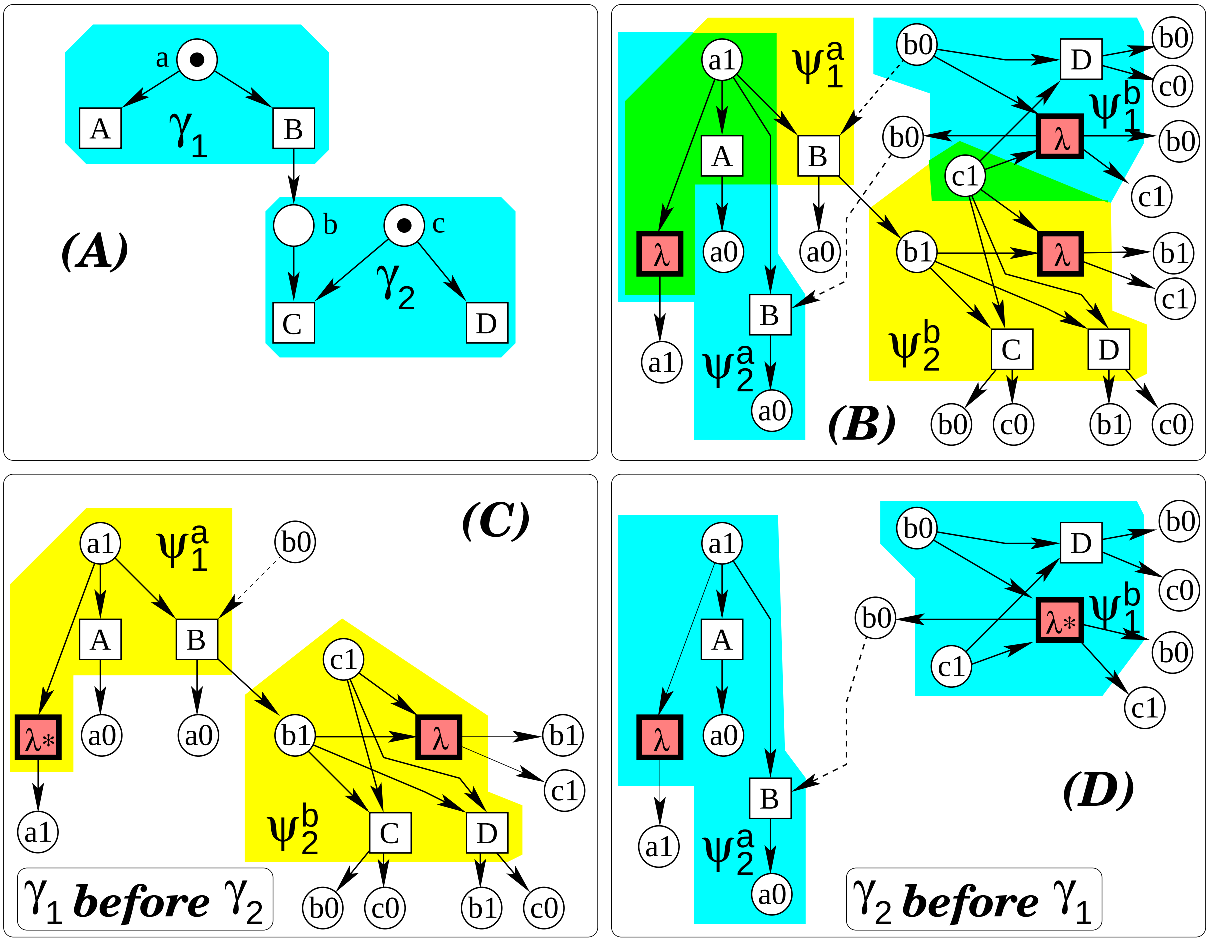

Figure 7: The non-determinism between clusters

3 Policy Directed Unfoldings

3.1 Ambiguity in Tiles and Non-Determinism Between Clusters

Clusters are pairwise disjoint by construction; this is not the case for tiles in general, as Figure 7 shows. There, cluster γ_2 has two different local markings in which it can fire transitions: the initial one shown in the figure, and the one after a firing of B and with a token on b. These two situations are reflected by two different tiles, ψ_1^b and ψ_2^b, both of which contain the condition c_1. In fact, the selection between ψ_1^b and ψ_2^b is done not by a choice *within* a cluster but rather by the *order* in which the two clusters γ_1 and γ_2 take their turns: if γ_1 is considered first, the process will contain ψ_1^a and ψ_2^b (Part (C) of Figure 7); if the possibilities of γ_2 are explored first, and *then* γ_1, the process consists of ψ_1^b and ψ_2^a, as shown in Part (D) of Figure 7. Note that choosing the event labeled "$\lambda*$" in (C) leads to the initial situation in (D), and vice versa; so under both orderings, the (essentially) same runs remain possible provided the "right" choices are made; however, in general, the probabilities will change.

So the fact that the tiles in (B) are not disjoint is only one aspect of a more fundamental problem: different clusters access the same place p in a non-deterministic order. Since only one cluster *takes* tokens from p, the other clusters have no direct influence, so we may assume no two accesses are simultaneous; the execution of a γ-step may also change the marking of one of its input places p, but since γ is the only cluster to which p belongs, no other cluster's choice depends on p.

3.2 ND-unfolding

We will start by a non-deterministic unfolding algorithm that constructs processes as defined in Definition 6. Intuitively, the procedure is the following: The initial marking is represented by a set of conditions b, such that $\pi(\mathsf{b})$ is a place and $\mu(\mathsf{b})$ the number of tokens present. Now, proceed inductively as follows: Choose a cluster γ and determine the set of γ-steps enabled under γ's current local marking (i.e., the global marking M restricted to the places p of γ). For each of these steps δ, create an event e (i.e. $\beta(\mathsf{e}) = \delta$); draw an $\mathbb{F}$-arc to e from all b in the initial cut such that $\pi(\mathsf{b})$ belongs to γ *or is an output place of* δ; put a copy b' of each pre-condition b of e behind e, i.e. with an $\mathbb{F}$-arc from e to b'; and set $\mu(\mathsf{b}')$ equal to $\mu(\mathsf{b})$ plus/minus the effect of δ. Further, for all post-places p of $\beta(\mathsf{e})$, draw an $\overline{\mathsf{F}}$-arc from the condition b that represents p in the current cut $\mathbf{c}$ to e, and a $\mathbb{F}$-arc from e to a new condition b' with $\pi(\mathsf{b}) = \mathsf{p}$; again, $\mu(\mathsf{b}')$ is determined by adding the effect of δ (which can only be positive here) to $\mu(\mathsf{b})$. Repeat the above for the newly obtained $\sqsubseteq$-maximal cuts. More formally:

Definition 8. (ND-unfolding) *Let* $\mathcal{N} = (\mathcal{P}, \mathcal{T}, \mathsf{F}, \mathsf{M}_0)$ *a Petri Net, with* $\mathcal{P} = \{\mathsf{p}_i \mid i \in \mathcal{I}\}$, *and* $\mathsf{B}_0 = \{\mathsf{b}_i \mid i \in \mathcal{I}\}$ *a copy of* $\mathcal{P}$.

1. *Let* $\mathbf{c}_0 = \mathsf{B}_0$; *for* $i \in \mathcal{I}$, *set* $\pi_0(\mathsf{b}_i) \overset{\mathsf{def}}{=} \mathsf{p}_i$ *and* $\mu_0(\mathsf{b}_i) \overset{\mathsf{def}}{=} \mathsf{M}_0(\mathsf{p}_i)$. *Denote as* $\overline{\mathsf{N}}_0$ *the occurrence net* $\overline{\mathsf{N}}_0 \overset{\mathsf{def}}{=} (\mathsf{B}_0, \emptyset, \emptyset)$, *and as* β_0 *the "mapping"* $\beta_0 : \emptyset \to \mathfrak{S}_\Gamma(\mathcal{N})$.

2. *Let* $n \in \mathbb{N}$, *and let* $\overline{\mathsf{N}}_n = (\mathsf{B}_n, \mathsf{E}_n, \overline{\mathsf{F}}_n)$ *be given, as well as* π_n, μ_n, *and* β_n. *Choose any cluster* $\gamma \in \Gamma(\mathcal{N})$. *To* $\overline{\mathsf{N}}_n$, *append a tile corresponding to the cluster* γ, *in the following way:*

   (a) *Denote the set of* co-*sets* $\mathfrak{a} \subseteq \mathsf{B}_n$ *such that* $\pi_{|\mathfrak{a}} : \mathfrak{a} \to \mathbf{in}(\gamma) \cup \mathbf{out}(\gamma)$ *is bijective, as* $\tilde{\mathfrak{A}}_n$, *and set* $\mathfrak{A}_n \overset{\mathsf{def}}{=} \max_{\sqsubseteq}(\mathfrak{A}_n)$ *(a new* γ*-tile will be appended to every set in* $\mathfrak{A}_n$*.)*

   (b) *For every coset* $\mathfrak{a}$, *let* $\mathsf{M}_\mathfrak{a}$ *be the associated marking, i.e.* $\mathsf{M}_\mathfrak{a} \circ \pi = \mu \cdot 1\!\!1_\mathfrak{a}$. *For* $\mathfrak{a} \in \mathfrak{A}_n$, *define* $\overline{\mathfrak{a}} \overset{\mathsf{def}}{=} (\mathfrak{a} \cap \pi^{-1}\mathbf{in}(\gamma))$ *and* $\tilde{\mathfrak{a}} \overset{\mathsf{def}}{=} \mathfrak{a} \backslash \overline{\mathfrak{a}}$; *that is,* $\overline{\mathfrak{a}}$ *contains the conditions corresponding to active pre-places in* γ, *and* $\tilde{\mathfrak{a}}$ *the passive conditions reflecting the output of those steps. Let*

   $$\mathsf{Enabled}(\mathfrak{a}) \overset{\mathsf{def}}{=} \{\delta \in \mathfrak{S}(\gamma) \mid \mathsf{M}_\mathfrak{a} \xrightarrow{\delta}\} \quad \textit{and} \quad \mathsf{E}_\mathfrak{a} \overset{\mathsf{def}}{=} \{\mathsf{e}_\delta \mid \delta \in \mathsf{Enabled}(\mathfrak{a})\}.$$

   Extend β_n *to* $\mathsf{Enabled}\mathfrak{a}$, *i.e.* $\beta_{n+1|\mathsf{E}_n} \equiv \beta_n$, *and* $\beta_{n+1}(\mathsf{e}_\delta) \overset{\mathsf{def}}{=} \delta$ *for all* $\delta \in \mathsf{Enabled}\mathfrak{a}$. *Now, set* $\mathsf{E}_{n+1} \overset{\mathsf{def}}{=} \mathsf{E}_n \cup (\bigcup_{\mathfrak{a} \in \mathfrak{A}_n} \mathsf{E}_\mathfrak{a})$. *For every* $\delta \in \mathsf{Enabled}\mathfrak{a}$, *let* $\mathfrak{a}_\delta$ *be the coset of conditions from* $\mathfrak{a}$ *affected by* δ*:*

   $$\mathfrak{a}_\delta \quad \overset{\mathsf{def}}{=} \quad \{\mathsf{b} \in \tilde{\mathfrak{a}} \mid \exists \mathsf{t} \in \delta : \ \pi(\mathsf{b}) \in^\bullet \mathsf{t}^\bullet\}\,.$$

   Of course, every $\mathfrak{a}_\delta$ *contains* $\overline{\mathfrak{a}}$ *but not necessarily all of* $\mathfrak{a}$. *Let* $\mathsf{B}_{\mathfrak{a},\delta}$ *be a set disjoint from* $\mathsf{B}_n \cup \mathsf{E}_n$, *such that there exists a bijection* $\kappa_{\mathfrak{a},\delta} : \mathfrak{a}_\delta \to \mathsf{B}_{\mathfrak{a},\delta}$, *and such that* $\delta \neq \delta'$ *implies* $\mathsf{B}_{\mathfrak{a},\delta} \cap \mathsf{B}_{\mathfrak{a},\delta'} = \emptyset$. *Extend* B_n *to*

   $$\mathsf{B}_{n+1} \quad \overset{\mathsf{def}}{=} \quad \mathsf{B}_n \ \cup \bigcup_{\mathfrak{a} \in \mathfrak{A}_n} \left[\bigcup_{\delta \in \mathsf{Enabled}\mathfrak{a}} \mathsf{B}_{\mathfrak{a},\delta} \right].$$

Accordingly, extend $\overline{\mathsf{F}}$ *as follows: set*

$$\overline{\mathsf{F}}_\mathfrak{a} \overset{\mathsf{def}}{=} \bigcup_{\delta \in \mathsf{Enabled}\mathfrak{a}} [(\mathfrak{a}_\delta \ \times \ \{\mathsf{e}_\delta\}) \ \cup \ (\{\mathsf{e}_\delta\} \ \times \ \mathsf{B}_{\mathfrak{a},\delta})] \quad \textit{for } \mathfrak{a} \in \mathfrak{A}_n\textit{, and} \quad \overline{\mathsf{F}}_{n+1} \overset{\mathsf{def}}{=} \overline{\mathsf{F}}_n \bigcup_{\mathfrak{a} \in \mathfrak{A}_n} \overline{\mathsf{F}}_\mathfrak{a}.$$

Let $\pi_{n+1} : \mathsf{B}_{n+1} \to \mathcal{P}$ *and* $\mu_{n+1} : \mathsf{B}_{n+1} \to \mathbb{N}_0$ *be the following extensions of* π_n *and* μ_n*:*

$$\pi_{n+1}(\mathsf{b}) \quad \overset{\mathsf{def}}{=} \quad \begin{cases} \pi_n(\mathsf{b}) & : \quad \mathsf{b} \ \in \mathsf{B}_n \\ \pi_n\,(\mathsf{b}') & : \quad \kappa^{-1}_{\mathfrak{a},\delta}\,(\mathsf{b}) = \mathsf{b}' \in \mathsf{B}_n \end{cases}$$

$$\mu_{n+1}(\mathsf{b}) \quad \overset{\mathsf{def}}{=} \quad \begin{cases} \mu_n(\mathsf{b}) & : \quad \mathsf{b} \in \mathsf{B}_n \\ \left[\mu_n\,(\mathsf{b}') - \langle \pi_n(\mathsf{b}')^{\odot}, \beta(\mathsf{e})\rangle\right] \ + \langle {}^{\odot}\pi_n(\mathsf{b}'), \beta(\mathsf{e})\rangle & : \quad \kappa^{-1}_{\mathfrak{a},\delta}\,(\mathsf{b}) = \mathsf{b}' \in \mathsf{B}_n \end{cases}$$

Set

$$\mathsf{B}_\star \overset{\mathsf{def}}{=} \bigcup_{n\in\mathbb{N}} \mathsf{B}_n,\ \mathsf{E}_\star \overset{\mathsf{def}}{=} \bigcup_{n\in\mathbb{N}} \mathsf{E}_n,\ \overline{\mathsf{F}}_\star \overset{\mathsf{def}}{=} \bigcup_{n\in\mathbb{N}} \overline{\mathsf{F}}_n.$$

Let $\pi_\star : \mathsf{B}_\star \to \mathcal{P}$*,* $\mu_\star : \mathsf{B}_\star \to \mathbb{N}_0$*, and* $\beta_\star : \mathsf{B}_\star \to \mathfrak{S}_\Gamma(\mathcal{N})$ *be the colimits, i.e. the unique mappings such that* $\pi_{\star|\mathsf{B}_n} \equiv \pi_n$*,* $\mu_{\star|\mathsf{B}_n} \equiv \mu_n$*, and* $\beta_{\star|\mathsf{B}_n} \equiv \beta_n$ *for all* n*. Then* $\Pi_\star = (\overline{\mathsf{N}}_\star, \pi_\star, \beta_\star, \mu_\star)$ *is an* ND-unfolding *of* $\mathcal{N}$*.*

We remark that the construction extends, in each round, *every* branch of the process by a tile associated to the cluster γ selected there . In fact, every maximal configuration *after* the nth round contains exactly one event per cluster of $\mathcal{N}$ added *during* the n-th round, since even the choice of inactivity in a cluster results in an event, labeled λ. The crucial step is number 2: Because of the non-determinism in the choice of γ in the above, the ND-unfolding is not unique. In fact, note first that *stuttering* is possible: suppose that in Step 2 of Definition 8 the same cluster γ is selected consistently in each round, and that only the empty step of γ is firable in M_0. Then the algorithm of Definition 8 yields an infinite branch of copies of the same tile, corresponding to γ under M_0, and all events labeled λ, and no other branch will be developed. The same effect occurs for any *loop*: Suppose transition t is only connected to place p, i.e. ${}^\bullet\mathsf{t} = \{\mathsf{p}\} = \mathsf{t}^\bullet$, and p is sufficiently marked under M_0. Then t can fire an infinite number of times without changing the marking. So, even to exclude the empty step would not be sufficient. The clusters of $\mathcal{N}$ act as local players, and the rules according to which they take their turns crucially influence the results, see below.

Full unfolding. First, we note that the unique maximal ND-unfolding can be obtained by forcing the algorithm to consider, in Step 2, first *all* clusters simultaneously, compute the corresponding tiles, and append them *together*; compare (B) in Figure 7. In fact, let Π_n be the process thus obtained by fully exploring all clusters in each round up to and including round n; then Π_{n_1} is a prefix[4] of Π_{n_2} for all $n_1 < n_2$. The family $(\Pi_n)_{n\in\mathbb{N}}$ then yields, as $n \to \infty$, a limit process $\hat{\Pi} = (\hat{\overline{\mathsf{N}}}, \hat{\pi}, \hat{\beta}, \hat{\mu})$, $\hat{\overline{\mathsf{N}}} = (\hat{\mathsf{B}}, \hat{\mathsf{E}}, \hat{\overline{\mathsf{F}}})$; call $\hat{\Pi}$ the **full unfolding** of $\mathcal{N}$. By construction, $\hat{\Pi}$ is unique up to an isomorphism of labeled graphs.

Remark 1. *Recall that the processes considered here all evolve in* logical *time, with no external clock to measure the evolution. However, it will be useful to have a set of* temporal constants*; we will use for this the n-th prefixes* Π_n *used above in introducing* $\hat{\Pi}$*.*

The full unfolding – more precisely, the space of its configurations – will be used, in the "probabilistic part" of this paper, as the space of *all* behaviors under *any* control. Now, as we saw in the context of Figure 7, *selecting* the clusters to be explored in the n-th round rather than exploring all clusters simultaneously, yields interesting subnets of the full unfolding.

3.3 Designs

Assume that clusters are inspected in the order given by some infinite sequence $\mathbf{d} = \mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3, \ldots$, where every $\mathbf{d}_i$ stands for the choice of some cluster γ of $\mathcal{N}$. Following the control theoretic terminology (e.g. [7]), we will speak of *designs* for such sequences. In the n-th action, the cluster γ_n given by $\mathbf{d}_n$ chooses one γ_n-step δ enabled in the current local marking on γ_n, and fires δ, thus possibly (note that δ can be λ) changing the marking on γ_n and its neighboring clusters. Then, γ_{n+1} will take its turn, in the new situation after γ_n's action, and so forth; call the resulting partial directed unfolding $\Pi_{\mathbf{d}}$. We note that for *fixed* $\mathbf{d}$, tiles in $\Pi_{\mathbf{d}}$ are pairwise disjoint.

Intuitively, the order of two independent clusters neighboring one another in $\mathbf{d}$ may be interchanged in the unfolding process without changing $\Pi_{\mathbf{d}}$. To formalize this, consider (Mazurkiewicz) traces of designs: denote the set of *infinite* (*arbitrary*) words over alphabet $\mathcal{C}$ as $\mathcal{C}^\infty$ ($\mathcal{C}^\omega$); and, for $\mathbf{d}, \mathbf{d}' \in \Gamma^\omega$,

$$\mathbf{d} \rightleftharpoons \mathbf{d}' \quad \overset{\mathsf{def}}{\Longleftrightarrow} \quad \exists n \in \mathbb{N}, \underline{\mathbf{d}} \in \Gamma^n,\ \overline{\mathbf{d}} \in \Gamma^\omega,\ \gamma, \gamma' \in \Gamma :\ \gamma \mathbb{I} \gamma' \wedge \left\{ \begin{array}{ccc} \mathbf{d} & = & \underline{\mathbf{d}}\gamma\gamma'\overline{\mathbf{d}} \\ \wedge\ \mathbf{d}' & = & \underline{\mathbf{d}}\gamma'\gamma\overline{\mathbf{d}} \end{array} \right\}.$$

[4] by abuse of terminology, we use the term *prefix* both for processes *and* the corresponding sets that span occurrence (sub)nets.

Write $\mathbf{d} \sim_{(\rightleftharpoons)} \mathbf{d}'$ iff *(i)* $\mathbf{d} = \mathbf{d}'$, or *(ii)* there exist $\mathbf{d}_1, \ldots, \mathbf{d}_k \in \Gamma^\omega$ such that $\mathbf{d} = \mathbf{d}_1 \rightleftharpoons \ldots \rightleftharpoons \mathbf{d}_n = \mathbf{d}'$. Obviously, $\sim_{(\rightleftharpoons)}$ is an equivalence relation; denote the $\sim_{(\rightleftharpoons)}$-class (the *trace*) of $\mathbf{d}$ as $[\mathbf{d}]$.

Theorem 1. $\mathbf{d} \sim_{(\rightleftharpoons)} \mathbf{d}'$ *implies that the unfoldings* $\Pi_{\mathbf{d}}$ *and* $\Pi'_{\mathbf{d}}$ *that are obtained choosing* $\mathbf{d}$ *and* $\mathbf{d}'$*, respectively, are identical processes up to an isomorphism of labeled graphs.*

Proof: It suffices to consider the case $\mathbf{d} \rightleftharpoons \mathbf{d}'$. Let n be the unique index such that $\mathbf{d}_n = \mathbf{d}'_{n+1}$ and $\mathbf{d}'_n = \mathbf{d}_{n+1}$. The processes Π_{n-1} and Π'_{n-1} obtained under the common prefix of $\mathbf{d}$ and $\mathbf{d}'$ have isomorphic occurrence nets $\overline{\mathsf{N}}_{n-1}$ and $\overline{\mathsf{N}}'_{n-1}$, and the isomorphism commutes, with π_{n-1}, μ_{n-1}, and β_{n-1} in both versions. The extensions added for the the clusters $\gamma \overset{\mathsf{def}}{=} \mathbf{d}_n$ and $\gamma' \overset{\mathsf{def}}{=} \mathbf{d}'_n$ are strictly disjoint since $\gamma \mathbb{I} \gamma'$, hence the nets $\overline{\mathsf{N}}_{n+1}$ obtained under $\mathbf{d}$ and $\mathbf{d}'$ are isomorphic, and the isomorphism commutes with the respective versions of π_{n+1}, μ_{n+1}, and β_{n+1}. Since $\mathbf{d}_k = \mathbf{d}'_k$ for all $k \geqslant n+2$, the claim follows by induction. □

So, it is possible to "parallelize" the construction of the unfolding using independence of distant clusters; for the branching process semantics, parallelization has been studied by Heljanko, Khomenko and Koutny in the forthcoming paper [29], using a different idea. Reviewing the proof of Theorem 1, we conclude that for any $\mathbb{I}$-clique $\mathcal{X} \subseteq \Gamma$, all clusters of $\mathcal{X}$ may be considered *simultaneously* in Step 2 of the algorithm in Definition 8, yielding the same unfolding (up to an isomorphism of labeled graphs) as any ordering of $\mathcal{X}$. This leads to the idea behind *cluster nets,* which we introduce next.

3.4 Cluster Net: Distributed Control for the Unfolding Algorithm

Theorem 1 tells us that the unfolding depends on $\mathbf{d}$ only via its *Mazurkiewicz trace* $[\mathbf{d}]$. Thus far we considered $\mathbf{d}$ as given; this implies the existence, at least in theory, of a global scheduler ordering the actions of all clusters of $\mathcal{N}$. This assumption is generally made in the literature on control for Petri nets (see for example [28, 39]); compare the *scheduler-luck* games (Dolev, Israeli and Moran [16]; see also [40]). We will show here that the scheduling itself can be done in a distributed way with a maximum of parallelization, using an abstract Petri Net. This *cluster net* $\mathcal{N}^\Gamma$ is not itself included in the structure of $\mathcal{N}$, which remains unchanged; rather, it provides a formal model producing all parallelized policies possible on $\mathcal{N}$. Thus, we can use a unifying model on several levels; in particular, the probability measures we will introduce below for the local *choice inside a cluster* can also be used on the higher level to schedule the global order in which clusters act. For this, we define:

Definition 9. *Let* $\mathcal{N}$ *be a Petri net with cluster set* $\Gamma = \Gamma(\mathcal{N})$, $\mathcal{P}^\Gamma \overset{\mathsf{def}}{=} \mathcal{P}$, *and*

$$\mathsf{F}^\Gamma \overset{\mathsf{def}}{=} \{(\gamma, \mathsf{p}), (\mathsf{p}, \gamma) \mid \mathsf{p} \in \gamma \ \wedge \ \exists \mathsf{t} \in \gamma : \ \mathsf{t} \ \mathsf{F} \ \mathsf{p}\}.$$

With $\mathcal{W}^\Gamma(\mathsf{x}, \mathsf{x}') = \mathbb{1}_{(\mathsf{F}^\Gamma)}(\mathsf{x}, \mathsf{x}')$*, call* $\mathcal{N}^\Gamma \overset{\mathsf{def}}{=} (\mathcal{P}^\Gamma, \Gamma, \mathcal{W}^\Gamma, \mathsf{M}^\Gamma)$ *the* cluster net *of* $\mathcal{N}$.

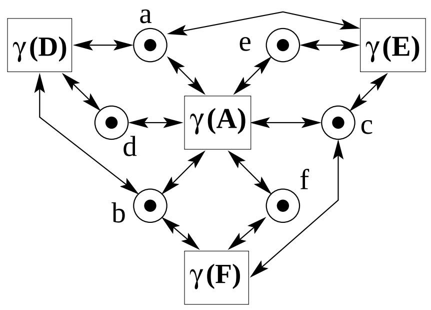

Figure 8: Cluster net for Fig. 3 and 6; two-sided arrows indicate the symmetric arc relation F^Γ

In words, $\mathcal{N}^\Gamma$ has one transition t_γ for every cluster γ of $\mathcal{N}$, and one copy p_γ of every place p from $\mathcal{N}$; p_γ is either **(i)** connected to t_γ by an two-way arc with weight one if $\mathsf{p} \in (\gamma \cup \mathbf{out}(\gamma))$, or **(ii)** no arc at all otherwise. Every p_γ contains one token, implementing mutual exclusion for actions of more than one neighboring cluster (including $\gamma(\mathsf{p})$). Since all arcs of $\mathcal{N}^\Gamma$ go in both directions, all transitions are *individually* enabled initially, and M^Γ is reproduced by every firing, so M^Γ is the only reachable marking in $\mathcal{N}^\Gamma$. One also deduces immediately that, if $\mathcal{N}$ is connected, $\mathcal{N}^\Gamma$ consists of a single cluster; in non-connected nets, we may simply treat each connected component separately, so assume without loss of generality that $\mathcal{N}^\Gamma$ is a single cluster.

Any design over $\mathcal{N}$ is a single transition firing sequence of $\mathcal{N}^{\Gamma}$. The enabled *steps* δ of $\mathcal{N}^{\Gamma}$ correspond 1-1 to the $\mathbb{I}$-cliques of clusters of $\mathcal{N}$; denote the set of these cliques as $\mathfrak{D} = \mathfrak{D}(\mathcal{N})$. Call every sequence $\Delta = (\Delta^{\Gamma}_{n})_{n \in \mathbb{N}}$ of cluster sets from $\mathcal{N}$ such that Δ represents a step firing sequence of $\mathcal{N}^{\Gamma}$ a **policy** for $\mathcal{N}$; policies thus generalize designs.

Definition 10. (Policy-directed unfolding) *Let* $\mathcal{N} = (\mathcal{P}, \mathcal{T}, \mathsf{F}, \mathsf{M}_0)$ *a Petri Net, and let* $\Delta = (\Delta^{\Gamma}_{n})_{n \in \mathbb{N}}$ *a policy for* $\mathcal{N}$. *Apply the construction of Definition 8, but with the following modification: in the* n*-th round, append one tile for each cluster* $\gamma \in \Delta^{\Gamma}_{n}(\gamma)$. *Set* $\mathsf{B}_{\Delta} \overset{\mathsf{def}}{=} \bigcup_{n \in \mathbb{N}} \mathsf{B}_n$, $\mathsf{E}_{\Delta} \overset{\mathsf{def}}{=} \bigcup_{n \in \mathbb{N}} \mathsf{E}_n$, *and* $\overline{\mathsf{F}}_{\Delta} \overset{\mathsf{def}}{=} \bigcup_{n \in \mathbb{N}} \overline{\mathsf{F}}_n$, *and let* $\pi_{\Delta} : \mathsf{B}_{\Delta} \to \mathcal{P}$, $\mu_{\Delta} : \mathsf{B}_{\Delta} \to \mathbb{N}_0$, *and* $\beta_{\Delta} : \mathsf{B}_{\Delta} \to \mathfrak{S}_{\Gamma}(\mathcal{N})$ *be the respective colimit mappings as above. Then* $\Pi_{\Delta} = (\overline{\mathsf{N}}_{\Delta}, \pi_{\Delta}, \beta_{\Delta}, \mu_{\Delta})$ *is the* Δ-unfolding *of* $\mathcal{N}$.

A prefix of the unfolding, for the net from Fig. 3, corresponding to $(\{\gamma_1\}\{\gamma_2, \gamma_3, \gamma_4\})^{\omega}$, is shown in Figure 6, omitting all $\emptyset$-labeled events for clarity of the drawing.

We have now completed the non-probabilistic part of the model.

4 The Probabilistic Model

Using the unfolding studied above, we will now construct the measurable space for runs and probability measures on it; the presentation uses standard terminology of probability theory, as used, e.g., in [12].

4.1 The Probability Space

Our probabilistic model needs to measure sets of *runs*. As we saw in the last section, the run realized by $\mathcal{N}$ is determined by a policy Δ and the subsequent choices of steps in the clusters selected by Δ.

The Policy Space: Let Ω_1 be the set of possible policies, and $\mathbb{P}^1$ a probability on the $\mathbb{I}$-cliques of $\mathcal{N}^{\Gamma}$ (see below for examples); let Δ_n be i.i.d., and for $\mathcal{X} \subseteq \Gamma$, let $\mathbb{P}^1\{\mathcal{X}\}$ be the probability that $\Delta_1 = \mathcal{X}$. So we identify ω_1 with the unique policy Δ it generates; the associated filtration is $(\mathcal{F}^1_n)_{n \in \mathbb{N}}$, where $\mathcal{F}^1_n$ is the σ-algebra generated by the random variables $\Delta_1, \ldots, \Delta_n$.

As we saw above, every policy Δ is associated to a unique unfolding $\Pi = \Pi_{\Delta}$ by virtue of Definition 10. Now, consider the set of clusters $\Delta_n = \Delta_n(\omega_1) \subseteq \Gamma$; by assumption, Δ is a clique of $\mathbb{I}$. Thus all $\gamma \in \Delta_n$ can choose their step independently of one another, given the local marking that has been reached after all decisions in $\Delta_1, ..., \Delta_{n-1}$.

The Choice Space: While the first component Ω_1 of the joint probability space is thus associated to the selection of a *policy*, identify Ω_2 with the sample space of the subsequent choices of the clusters. Every choice of γ is a vector $\delta^{(\gamma,n)} = (\delta^{(\gamma,n)}_{\mathsf{M}})_{\mathsf{M} \in \mathfrak{M}(\mathcal{P}_{\gamma})}$ of random variables taking values in the set of steps of γ; that is, a random vector whose components are indexed by the possible markings that γ may find on $\mathcal{P}_{\gamma}$ when making its n-th choice. Further, $(\delta^{(\gamma,n)}_{\mathsf{M}})_{\gamma \in \Gamma, n \in \mathbb{N}}$ is a doubly indexed field of independent variables, i.e. the choices on different clusters, and the different choices on the same cluster, are made independently of one another.

Starting from $\mathbf{c}_0$, one obtains a unique run ω using a fixed $(\omega_1, \omega_2) \in \Omega_1 \times \Omega_2$; so we will identify ω with (ω_1, ω_2) and Ω with $\Omega_1 \times \Omega_2$.

Cluster measures. For $\gamma \in \Gamma(\mathcal{N})$ and a local marking $\mathsf{M}\gamma \in \mathfrak{M}(\mathcal{P}_{\gamma})$, write $\mathsf{Enabled}(\mathsf{M}, \gamma)$ for the set of γ-steps enabled under M_{γ}. A *choice measure family* or CMF for γ is a transition probability function $\mathbb{P}^{\gamma} = (\mathbb{P}^{\gamma}_{\mathsf{M}})_{\mathsf{M} \in \mathfrak{M}(\mathbf{in}(\gamma))}$ such that, for a given $\mathsf{M} \in \mathfrak{M}(\mathbf{in}(\gamma))$, $\mathbb{P}^{\gamma}_{\mathsf{M}}$ is a probability on $\mathsf{Enabled}(\mathsf{M}, \gamma)$. A *cluster measure family* is a family $(\mathbb{P}^{\gamma})_{\gamma \in \Gamma(\mathcal{N})}$ of CMFs. Two important examples for construction of cluster measures will be given in Subsection 4.2 below.

Tile measures. Now, any cluster measure family induces a family of *tile measures*: for $\mathsf{e} \in \mathsf{E}$ and $\psi = \psi(\mathsf{e})$, let $\gamma \overset{\mathsf{def}}{=} \gamma(\beta(\mathsf{e}))$, and set $\mathsf{M}(\mathsf{e}) \overset{\mathsf{def}}{=} \mu \cdot \mathbb{1}_{\circ \mathsf{e}}$ and $\mathbb{P}^{\psi}(\mathsf{e}) \overset{\mathsf{def}}{=} \mathbb{P}^{\gamma}_{\mathsf{M}(\mathsf{e})}(\mathsf{e})$.

Filtrations. The following construction of filtrations $(\mathcal{F}_{\mathcal{U}})_{\mathsf{Pref}})$ for Ω follows the one introduced in [8], with a probability space containing all *runs* and filtered by increasing *prefixes*; compare Völzer's [40] *cones.*

Let $\mathcal{N} = (\mathcal{P}, \mathcal{T}, \mathcal{W}, \mathsf{M})$ be a Petri Net, and $\hat{\Pi} = (\hat{\overline{\mathsf{N}}}, \hat{\pi}, \hat{\beta}, \hat{\mu})$ the *full cluster unfolding* of $\mathcal{N}$; here, $\hat{\overline{\mathsf{N}}} = (\mathsf{B}, \mathsf{E}, \overline{\mathsf{F}})$, with initial cut $\mathbf{c}_0$. Recall that all processes are *prefixes* of $\hat{\Pi}$; we denote a prefix generically as $\mathcal{U}$, and the set of prefixes of $\mathcal{N}$ is $\mathsf{Pref}(\mathcal{N})$. For a run $\omega \in \Omega$ and a prefix $\mathcal{U}$, define the *$\mathcal{U}$-segment of* ω as the configuration $\omega_{\mathcal{U}} \overset{\mathsf{def}}{=} \omega \cap \mathcal{U}$. For every fixed $\mathcal{U}$, the relation $\sim_{\mathcal{U}}$ given by

$$\omega \sim_{\mathcal{U}} \omega' \quad \overset{\mathsf{def}}{\Longleftrightarrow} \quad \omega_{\mathcal{U}} = \omega'_{\mathcal{U}}, \tag{7}$$

is an equivalence on Ω; its extension to $\mathbb{C}$ is an equivalence as well. Let $\mathcal{F}_{\mathcal{U}}$ be the σ-algebra given by:

$$\mathcal{A} \in \mathcal{F}_{\mathcal{U}} \quad \overset{\mathsf{def}}{\Longleftrightarrow} \quad \left[\left(\begin{array}{c} \omega \in \mathcal{A} \\ \omega' \sim_{\mathcal{U}} \omega \end{array} \right) \Rightarrow \omega' \in \mathcal{A} \right] \tag{8}$$

Thus $(\mathcal{F}_{\mathcal{U}})_{\mathcal{U} \in \mathsf{Pref}}$ is a filtration over Ω. For a node x, set $\mathcal{A}_{\mathsf{x}} \overset{\mathsf{def}}{=} \{\omega \mid \mathsf{x} \in \omega\}$; by extension, for a set $\mathcal{X}$ of nodes, write $\mathcal{A}_{\mathcal{X}} \overset{\mathsf{def}}{=} \{\omega \mid \mathcal{X} \subseteq \omega\}$. If no confusion can occur, we just write x $(\mathcal{X})$ for the set $\mathcal{A}_{\mathsf{x}}$ $(\mathcal{A}_{\mathcal{X}})$. For $\mathbf{c} \in \mathfrak{C}(\overline{\mathsf{N}})$, let $\overline{\pi}_{\mathbf{c}}$ be the mapping that associates to every place p its unique representative in $\mathbf{c}$, i.e.

$$\pi^{-1}(\{\mathsf{p}\}) \ \cap \ \mathbf{c} \quad = \quad \{\overline{\pi}_{\mathbf{c}}(\mathsf{p})\}.$$

For a fixed prefix $\mathcal{U}$, define $(\omega \in \Omega)$: $\mathbf{C}_{\mathcal{U}}(\omega) \overset{\mathsf{def}}{=} \omega_{\mathcal{U}}$. Now, if $\mathbf{C}_{\mathcal{U}}$ is finite, $\mathbf{c}_{\mathcal{U}}(\omega) \overset{\mathsf{def}}{=} \max(\mathbf{C}_{\mathcal{U}}(\omega))$ exists and is a cut; in that case, the *marking at time* $\mathcal{U}$ is the random variable $\mathsf{M}_{\mathcal{U}}(\omega) \overset{\mathsf{def}}{=} \mu \circ \overline{\pi}_{\mathbf{c}_{\mathcal{U}}(\omega)}$.

Joint Probability. We construct inductively a probability $\mathbb{P}$ on Ω, adapted to $(\mathcal{F}_{\mathcal{U}})_{\mathcal{U} \in \mathbb{N}}$. Initialize with $\mathbb{P}_0(\mathbf{c} \in \omega) = 1$. For $n \in \mathbb{N}$, let $\mathbf{C}_n = \mathbf{C}_n(\omega)$, $\mathbf{c}_n = \mathbf{c}_n(\omega)$ and M_n as above. That is, in the section $\underline{\mathfrak{n}}(\omega_1)$ of prefix $\underline{\mathfrak{n}}$ given by $\Delta_1(\omega_1), \ldots, \Delta_n(\omega_1)$, the cluster choices governed by ω_2 have let to the configuration $\mathbf{C}_n$, ending in condition cut $\mathbf{c}_n$ which represents marking M_n. Further, let ψ be a tile such that $\mathbf{in}(\psi) \subseteq \mathbf{c}_n$, and e an event from ψ; denote as δ the step $\beta(\mathsf{e})$, and as γ the cluster corresponding to ψ. Then, define inductively a family $\mathbf{C}_n$ of configuration- valued random variables, and the associated probability measures $\mathbb{P}_n$, such that $\mathbb{P}_0$ and $\mathbf{C}_0$ satisfy $\mathbb{P}_0(\mathbf{C}_0 = \mathbf{c}_0) = 1$, and

$$\begin{aligned} \mathbb{P}_{n+1}(\mathsf{e} \in \mathbf{C}_{n+1} \mid \mathbf{C}_n) \quad &= \quad \mathbb{P}^1(\gamma \in \Delta_{n+1}) \cdot \mathbb{P}^2_{\gamma}(\delta_n \mid \mathsf{M}_n) \\ &= \quad \mathbb{P}^1(\gamma \in \Delta_1 \ ) \cdot \mathbb{P}^{\gamma}_{\mathsf{M}_n}(\mathsf{e} \in \mathbf{C}_{n+1}). \end{aligned} \tag{9}$$

Remark 2. *The above definitions have important consequences.*

1. *The recursion (9) yields a limit probability that we denote by* $\mathbb{P}$*;* $\mathbb{P}$ *is unique. To see this, we use the Kolmogorov Extension Theorem (see for example [12]) with essentially the same argument as in [8]: Take the increasing sequence* $(\underline{\mathfrak{n}})_{n \in \mathbb{N}}$ *of finite prefixes, given by Definition 10. Now, setting* $\mathbb{P}_0(\mathbf{c}_0) \overset{\mathsf{def}}{=} 1$ *one obtains inductively from (9) a consistent family of marginal probability measures* $\mathbb{P}_n$*; by Kolmogorov's Extension Theorem, the projective limit for this family exists (on* $\Omega_{\mathcal{N}} = \bigcup_{n \in \mathbb{N}} \underline{\mathfrak{n}}$*), and is unique. We denote this probability as* $\mathbb{P}$*.*

2. *Further, one obtains a "random prefix"* $\mathcal{U}_n = \mathcal{U}(\Delta_1, \ldots, \Delta_n)$ *in the following way: set* $\mathcal{U}_0 \overset{\mathsf{def}}{=} \mathbf{c}_0$*; inductively add to* $\mathcal{U}_n$ *all events corresponding to a step enabled in cluster* Δ_n *under a marking* $\mathsf{M}(\mathcal{U}_n(\omega))$*, and the post-conditions of that event. For all* n*, the configuration* $\mathbf{C}_n$ *is finite and thus yields a unique marking* $\mathsf{M}_n \overset{\mathsf{def}}{=} \mathsf{M}_{\mathbf{C}_n}$ *as above. So* M_n *is a random variable in* $(\Omega, \mathcal{F}_n, \mathbb{P}_n)$*.*

3. *From the recursive construction of* $\mathbb{P}$ *via (9), it comes as no surprise to see that the resulting process is Markov; below, we will introduce the appropriate notion of* stopping time *and prove even a* strong Markov property.

4.2 Cluster measures

Consider a single cluster γ of $\mathcal{N}$ and the γ-steps enabled under the restriction $\mathsf{M}_{|\gamma}$ of a marking M to γ. We postulate that, to be coherent with Petri net dynamics, a probability measure for the choice of the step to be fired should depend in a functional way on $\mathsf{M}_{|\gamma}$. As we saw earlier, there is no *choice* involved in the *creation* of tokens: a step that fires is assured to produce its tokens on its post-places. By contrast, it is at the *beginning* of a step that choices are made, i.e. the enabled steps compete for the *incoming* tokens. Therefore, we have to equip with probabilities the input, and not the output of tokens: given the choice of a step, the number and places of tokens to be created is fully determined. In what follows, let $\tilde{\Omega} \overset{\mathsf{def}}{=} \mathfrak{S}(\gamma)$, let δ be the random element of $\mathfrak{S}(\gamma)$ to be selected, and $\mathcal{F}$ the σ-algebra generated by the sets $\mathcal{X}_{\mathsf{t},n} \overset{\mathsf{def}}{=} \{\omega \in \Omega \mid \delta(\mathsf{t}) \leqslant n\}$, where t runs over $\mathcal{T}_\gamma$ and n over $\mathbb{N}_0$.

In a cluster, all transitions influence one another by a direct conflict, or indirectly through a chain of conflicts. However, suppose two transitions $\mathsf{t}_1, \mathsf{t}_2 \in \mathcal{T}_\gamma$ are *structurally independent*, i.e. $({}^\bullet\mathsf{t}_1 \cup \mathsf{t}_1{}^\bullet) \cap {}^\bullet\mathsf{t}_2 = \emptyset$ and $({}^\bullet\mathsf{t}_2 \cup \mathsf{t}_2{}^\bullet) \cap {}^\bullet\mathsf{t}_1 = \emptyset$; denote this as "$\mathsf{t}_1$ **ind** t_2". Then the number of times t_1 fires should not depend on t_2, given the the behavior of the other transitions of γ. Of course, there will be some influence of t_2 on the possibilities of t_1's opponents; what we mean is that once the behavior of the rest of γ is fixed and known, the conditional laws for t_1 and t_2 are independent. To formalize this, we have to make a short excursion in *Markov Field* theory, see [15, 24, 31]. For the local marking M on γ, write (t_1 $\mathbf{dep}_\mathsf{M}$ t_2) if *(i)* $\mathsf{t}_1 \neq \mathsf{t}_2$, *(ii)* $\mathsf{M} \xrightarrow{\mathsf{t}_1}$ and $\mathsf{M} \xrightarrow{\mathsf{t}_2}$, and *(iii)* ${}^\bullet\mathsf{t}_1 \cap {}^\bullet\mathsf{t}_1 \neq \emptyset$; then $\mathcal{G} = \mathcal{G}^\gamma_\mathsf{M} \overset{\mathsf{def}}{=} ((\mathcal{T}_\gamma), \mathbf{dep}_\mathsf{M})$ is an undirected graph, in which two transitions are connected by an edge iff they are actively competing for tokens on some place. Note that a change of M changes the topology of $\mathcal{G}$: an edge from t_1 to t_2 is destroyed if one of the two transition is no longer enabled under the new marking; and edges are created from newly enabled transitions to their opponents. We are interested in the random vector $\xi^\mathcal{G} \overset{\mathsf{def}}{=} (\xi(\mathsf{t}))_{\mathsf{t}\in\mathcal{T}_\gamma}$, where ξ is some observable[5]; each component of ξ takes values in some common state space Ξ. Here, ξ will be related to the γ-step to be selected; ξ can be *identified* with the step in the second example below, and will be *converted* into the step in the first example.

Now, for a subset $\mathcal{X} \subseteq \mathcal{T}_\gamma$, define the **boundary** of $\mathcal{X}$ as

$$\partial\mathcal{X} \overset{\mathsf{def}}{=} \{\mathsf{t} \mid \exists\, \mathsf{t}' \in (\mathcal{T}_\gamma \backslash \mathcal{X}) : \ \mathsf{t}\ \mathbf{dep}\ \mathsf{t}'\}\,.$$

$\xi^\mathcal{G}$ is called (see [31]) a **Markov field** iff for all $\mathcal{X} \subseteq \mathcal{T}_\gamma$,

$$\mathbb{P}\left(\xi_{|\mathcal{X}} \mid \xi_{|(\mathcal{T}_\gamma \backslash \mathcal{X})}\right) = \mathbb{P}\left(\xi_\mathcal{X} \mid \xi_{|\partial\mathcal{X}}\right). \tag{10}$$

This requirement is natural in the light of Petri net semantics: the behaviors of two (sets of) transitions that are not close neighbors are independent, once the behavior of their close neighbors is known. So, Markov fields are a natural and important class of measures for cluster choices. It is well known (see [31]) that the following construction based on *potentials* yields Markov fields.

A *Gibbs potential* is a mapping D that associates to each subset $\mathcal{X}$ of $\mathcal{T}_\gamma$ and observable state ξ a number $D_\mathcal{X}(\xi) = D(\xi_{|\mathcal{X}})$, such that $D_\mathcal{X}(\xi) = 0$ unless $\mathcal{X}$ is a clique of $\mathcal{G}^\gamma_\mathsf{M}$. The *energy* U of ξ is

$$U(\xi) \overset{\mathsf{def}}{=} -\sum_{\mathcal{X} \subseteq \mathcal{T}_\gamma} D_\mathcal{X}(\xi); \tag{11}$$

obviously, the sum needs only be taken over the cliques of $\mathcal{G}^\gamma_\mathsf{M}$. The *partition function* is

$$Z^\gamma_\mathsf{M} \overset{\mathsf{def}}{=} \sum_{\xi\in\Xi} \mathsf{e}^{-U(\xi)}; \tag{12}$$

if Z^γ_M is finite, the *Gibbs measure* associated to the potential D is given by

$$\mathbb{P}^\gamma_\mathsf{M}(\xi_0) \overset{\mathsf{def}}{=} \frac{\mathsf{e}^{-U(\xi_0)}}{Z^\gamma_\mathsf{M}}. \tag{13}$$

For Petri nets, the variable we want to describe is the step δ selected by γ; δ is assumed to depend in a functional way on ξ, where the exact form of this dependency will change with the approach taken; see

[5] In the usual terminology of the literature on random and, in particular, Markov fields, the elements of the compound state space are called *configurations*; we do not follow this tradition here to avoid confusion with the configurations of concurrent processes.

the two examples below. What we are *really* interested in is the conditional probability that δ_0 is selected *under the condition that some* **admissible** *step*, i.e. enabled under M, is selected; that is, the probability

$$\mathbb{P}_{\mathsf{M}}\left(\delta = \delta_0 \mid \mathcal{X}_{\mathsf{Enabled(M)}}\right), \tag{14}$$

where $\mathcal{X}_{\mathsf{Enabled(M)}}$ denotes the event that the random δ is such that $\mathsf{M} \xrightarrow{\delta}$. In fact, no other outcomes for δ are admissible in the Petri net dynamics. Now, whenever

$$\mathbb{P}_{\mathsf{M}}(\mathcal{X}_{\mathsf{Enabled(M)}}) \quad > \quad 0, \tag{15}$$

the probability (14) is simply

$$\mathbb{P}_{\mathsf{M}}\left(\delta = \delta_0 \mid \mathcal{X}_{\mathsf{Enabled(M)}}\right) \quad = \quad \frac{\mathbb{P}_{\mathsf{M}}\left(\delta_0\right)}{\mathbb{P}_{\mathsf{M}}\left(\mathcal{X}_{\mathsf{Enabled(M)}}\right)}; \tag{16}$$

the non-degeneracy assumption (15) will be satisfied in both examples below.

4.2.1 Token Routing

Routing probabilities were used, e.g., in [8] and, as *coin flips*, in [40]. For every place $\mathsf{p} \in \mathcal{P}_\gamma$, introduce $\mathsf{M(p)}$ i.i.d. random variables taking values in $\mathsf{p}^\bullet \subseteq \mathcal{T}_\gamma$; that is, an i.i.d. router assigns every token on p under M to a post-transition of p.

First, however, we modify the structure of γ to account for *idling*: since we do not assume maximal step firing in general, we do not force ξ to assign every token for consumption. Formally, we add, for every place p of γ, a new transition $\mathsf{t_p}$ *"looped around* p*"*, i.e. ${}^\bullet\mathsf{t_p} = \mathsf{t_p}^\bullet = \{\mathsf{p}\}$, and $\mathcal{W}(\mathsf{p}, \mathsf{t_p}) = \mathcal{W}(\mathsf{t_p}, \mathsf{p}) = 1$,. These $\mathsf{t_p}$'s model *idling*, i.e. if a step $\delta \in \mathfrak{S}(\gamma)$ leaves k tokens on place p, we regard δ as containing k firings of $\mathsf{t_p}$. In this way, we only need to consider steps that use all tokens on $\mathbf{in}(\gamma)$; set

$$\hat{\mathcal{T}}_\gamma \overset{\mathsf{def}}{=} \mathcal{T}_\gamma \cup \{\mathsf{t_p} \mid \mathsf{p} \in \mathcal{P}_\gamma\}.$$

Then, fix routing probabilities π_{p} i.i.d. on the postset $\mathsf{p}^\bullet$ of p in $\hat{\mathcal{T}}$; that is, any token on p is routed to $\mathsf{t} \in \mathsf{p}^\bullet$ with probability $\pi_{\mathsf{p}}(\mathsf{t})$ independently of other tokens and other nodes. In the vector family $(\xi_{\mathsf{t}})_{\mathsf{t} \in \mathcal{T}_\gamma}$, each vector $(\xi_{\mathsf{t}}(\mathsf{p}))_{\mathsf{p} \in {}^\bullet\mathsf{t}}$ indicates, component by component, the random number of tokens on place p thus assigned to t. So, we will condition on the set where *offer equals demand*:

$$\mathsf{Enabled}^\xi \overset{\mathsf{def}}{=} \{\delta \in \mathfrak{M}(\mathcal{T}_\gamma) \mid \forall \mathsf{p} \in \mathbf{in}(\gamma),\ \mathsf{t} \in \mathsf{p}^\bullet :\ \xi_{\mathsf{t}}(\mathsf{p}) = \mathcal{W}(\mathsf{p}, \mathsf{t}) \cdot \delta(\mathsf{t})\}. \tag{17}$$

Clearly, for a fixed $\xi \in \Xi$, the set $\mathsf{Enabled}^\xi$ is either empty or a singleton; in the latter case, $\delta = \delta(\xi)$. Moreover, the definition (17) is local in the following sense: If $\mathsf{Enabled}^\xi$ is empty, then there exists a transition t of $\mathcal{G}_\gamma$ such that $\xi_{\mathsf{t}}(\mathsf{p}) \neq \mathcal{W}(\mathsf{p}, \mathsf{t})$. For every clique $\mathcal{X} \subseteq \hat{\mathcal{T}}_\gamma$ of $\mathcal{G}_\gamma$, set

$$\mathcal{A}(\mathcal{X}) \overset{\mathsf{def}}{=} \{\xi \in \Xi \mid \exists\, k \in \mathbb{N} :\ [\forall \mathsf{t} \in \mathcal{X}, \mathsf{p} \in {}^\bullet\mathsf{t} :\ \xi_{\mathsf{t}}(\mathsf{p}) = k \cdot \mathcal{W}(\mathsf{p}, \mathsf{t})]\};$$

note that all cliques of $\mathcal{G}_\gamma$ corresponds to transition sets having (at least) one common pre-place. Define the conditional (on $\mathsf{Enabled}^\xi$) potential as:

$$D^{\mathsf{M}}_{\mathcal{X}}(\xi) \overset{\mathsf{def}}{=} \mathbb{1}_{\mathcal{A}(\mathcal{X})}(\xi) \cdot \left[\sum_{\mathcal{P} \in {}^\bullet\mathsf{t}} \delta(\mathsf{t}) \cdot \ln(\nu_{\mathsf{p}}(\mathsf{t}))\right] + \mathbb{1}_{(\Xi \setminus \mathcal{A}(\mathcal{X}))}(\xi) \cdot \infty.$$

In fact, the only non-vanishing summands contributing to Z^γ_{M} are those coming from enabled steps. The Gibbs measure $\mathbb{P}^\gamma_{\mathsf{M}}$, associated to D^{M}, is exactly the conditional probability (14), since

$$\mathbb{P}_{\mathsf{M}}(\mathcal{X}_{\mathsf{Enabled(M)}}) \quad = \quad \frac{\sum_{\delta \in \mathsf{Enabled(M)}} \mathsf{e}^{-U(\delta)}}{\sum_{\delta \in \mathfrak{M}(\mathcal{T}_\gamma)} \mathsf{e}^{-U(\delta)}}.$$

In the example on the left hand side of Figure 4, with one token on each a, b and c, one has

$$\mathbb{P}^{\gamma}_{\mathsf{M}}\left(\delta = \mathbb{1}_{\{A\}}\right) \quad = \quad \frac{\nu_a(A)\cdot[\nu_b(\mathsf{t}_b)\cdot\nu_c(\mathsf{t}_c)+\nu_b(C)\cdot\nu_c(C)+\nu_b(D)\cdot\nu_c(D)]}{\left(\begin{array}{rcl} \nu_a(A) & \cdot & [\nu_b(\mathsf{t}_b)\cdot\nu_c(\mathsf{t}_c)+\nu_b(C)\cdot\nu_c(C)+\nu_b(D)\cdot\nu_c(D)] \\ +\nu_a(B) & \cdot & [\nu_b(B)\cdot\nu_c(\mathsf{t}_c)] \\ +\nu_a(\mathsf{t}_a) & \cdot & [\nu_b(\mathsf{t}_b)\cdot\nu_c(\mathsf{t}_c)+\nu_b(C)\cdot\nu_c(C)+\nu_b(D)\cdot\nu_c(D)] \end{array}\right)}.$$

That is, probability of agreement of all routers on the step $\mathbb{1}_{\{A\}}$ divided by the probability of agreement on *some* enabled step: *offer equals demand.*

4.2.2 Transition Coin Toss

In the second strategy, it is not the tokens that choose but the *transitions:* Assume every transition $\mathsf{t} \in \gamma$ chooses its firing degree $\delta(\mathsf{t})$ by a *"coin toss"* decision $\xi_{\mathsf{t}} \in \mathbb{N}_0$; then, discard the choices combinations that do not yield an enabled step. Here, we have *no* idling transitions, and we can identify $\delta = \xi$. For $\mathsf{t} \in \mathcal{T}_\gamma$ and $k \in \mathbb{N}$, denote as $\eta_{\mathsf{t}}(k)$ the probability that $\xi_{\mathsf{t}} = k$. For every clique $\mathcal{X} \subseteq \hat{\mathcal{T}}_\gamma$ of $\mathcal{G}_\gamma$, set

$$\mathcal{A}(\mathcal{X}) \quad \overset{\mathsf{def}}{=} \quad \{\xi \mid \forall\ \mathsf{p}\ \in\ {}^{\bullet}\mathcal{X} :\ \sum_{\mathsf{t}\in\mathsf{p}^{\bullet}\cap\mathcal{X}} \eta_{\mathsf{t}} \geqslant \xi_{\mathsf{t}}(\mathsf{p}) = k\cdot\mathcal{W}(\mathsf{p},\mathsf{t})\};$$

then,

$$D_{\mathcal{X}}(\delta) \quad \overset{\mathsf{def}}{=} \quad \mathbb{1}_{\mathcal{A}(\mathcal{X})}(\xi)\cdot\sum_{\mathsf{t}\in\mathcal{X}}\xi(\mathsf{t})\cdot\ln\left[\eta_{\mathsf{t}}(\xi(\mathsf{t}))\right] \ +\ \mathbb{1}_{\mathcal{A}(\mathcal{X})}(\xi)\cdot\infty.$$

In the context of Figure 4, this leads to

$$\mathbb{P}^{\gamma}_{\mathsf{M}}\left(\mathbb{1}_{\{A\}}\right) \quad = \quad \frac{\eta_A(1)\cdot[\eta_B(1)\cdot\eta_C(0)+\eta_B(0)\cdot\eta_C(1)+\eta_B(0)\cdot\eta_C(0)]}{\left(\begin{array}{rcl} \eta_A(1) & \cdot & [\eta_B(1)\cdot\eta_C(0)+\eta_B(0)\cdot\eta_C(1)+\eta_B(0)\cdot\eta_C(0)] \\ +\eta_A(0) & \cdot & [\eta_B(1)\cdot\eta_C(0)+\eta_B(0)\cdot\eta_C(1)+\eta_B(0)\cdot\eta_C(0)] \end{array}\right)}.$$

4.2.3 ... and more ?

The above list of possible cluster firing policies cannot be complete; it is meant only to give an idea of the variety of models that will fit into the framework we present here. Some of the above have been considered in the literature, although in less generality; in [1, 2], conflicts between individual transitions in GSPN are randomized using static priorities and transition weights: the probability of firing t is the weight of t, divided by the sum of the weights of *all* transitions in the *conflict set.* These conflict sets coincide with the *clusters* here, in the class of Petri Nets where both definitions are applicable (in particular, where all transitions have equal priority). There, single firing of transitions is required; so, steps of transitions with a common pre-place are not considered as enabled, and auto-concurrency of transitions is excluded. With the obvious generalization of the weight approach to steps, this randomization can be included in our list. However, the framework of [2] is in a *global time* setting, and the non-determinism between clusters is not resolved; so we can integrate only the idea of local weight randomization in our model, and *inside* clusters only.

Note further that the enabling rule used here, known as *step firing rule*, is not the only one used in the literature; two important other rules:

- The *single transition firing rule* allows only steps of the form $\delta_{\tilde{\mathsf{t}}}(\mathsf{t}) \overset{\mathsf{def}}{=} \mathbb{1}_{\{\tilde{\mathsf{t}}\}}(\mathsf{t})$ for some $\tilde{\mathsf{t}} \in \mathcal{T}$.
- Call a step $\delta \in \mathsf{Enabled}(\mathsf{M})$ *maximal* for M iff no further firing instance of any t can be added to δ without rendering (1) false for some $\mathsf{p} \in {}^{\bullet}\mathsf{t}$:

  $$\forall\ \mathsf{t} \in \mathcal{T}\ \exists\ \mathsf{p} \in {}^{\bullet}\mathsf{t} : \mathsf{M}(\mathsf{p}) < \langle \mathsf{p}^{\odot}, \delta\rangle + \mathcal{W}(\mathsf{p},\mathsf{t}). \qquad (18)$$

  Allowing only maximal steps yields the *maximal firing rule.*

A change of the *firing rule* e.g. to *single transition* firing, can be implemented by setting the probabilities of unwanted steps to zero; this will in general destroy the Markov field property since the constraints imposed by firing rules like the above are global for the cluster.

5 Stopping Times and Markov Property

The probability obtained from a choice measure family according to (9) randomizes the *local* choices only; however, the randomness of a non-sequential process is brought about not by the local choices alone, but also the ordering of the actions, see above.

5.1 Definition and Properties of Stopping Times

Let us have a closer look at the role of *time*. The standard notion of *stopping times* does not extend in a straightforward way to the processes here, since their defining propertyrequires a set $\mathbb{T}$ of temporal constants *external* to the evolution of the process. These constants will be the prefixes $\underline{\mathfrak{n}}$, $n \in \mathbb{N}$, introduced above; recall that $\underline{\mathfrak{n}}$ contains all configurations possible under any policy and any choices *up to and including the nth round.*

Definition 11. *Let $\mathcal{N}$ and $\underline{\mathfrak{n}}$ as above. Then a* stopping time *of $\mathcal{N}$ is a prefix τ satisfying*

$$\forall\, n \in \mathbb{N}: \ \{\omega \mid \omega_\tau \subseteq \underline{\mathfrak{n}}\} \quad \in \quad \mathcal{F}_{\underline{\mathfrak{n}}}. \tag{19}$$

Denote the set of stopping times as $\mathfrak{T}$; we will often write $\tau(\omega)$ for ω_τ.

Of course, if $\mathcal{N}$ is an S-set, the stopping times as defined coincide with the usual ones in linear times.

We now give a structural characterization of stopping times: stopping times are exactly those prefixes that *respect tiles*, in the following sense:

Definition 12. *A prefix $\mathcal{U}$ is* tile-respecting *iff for any tile ψ, $[\psi \ \cap \ \mathcal{U} \ \cap \ \mathsf{E}] \ \neq \ \emptyset$ implies $\psi \ \subseteq \ \mathcal{U}$.*

In words, the tile-respecting prefixes are those that contain all tiles with which they share an event; consequently, tile-respecting prefixes are *composed* of tiles.

Theorem 2. *A prefix τ of the full unfolding is a stopping time iff it is tile-respecting.*

Proof: Note first that all $\underline{\mathfrak{n}}$ are tile-respecting by construction. Suppose first that τ is a stopping time such that $\Psi_\tau \neq \emptyset$, where Ψ_τ is the set

$$\Psi_\tau \quad \overset{\mathsf{def}}{=} \quad \{\psi \mid \mathsf{E}(\psi) \overset{\mathsf{def}}{=} (\mathsf{E} \cap \tau \cap \psi) \neq \emptyset \ \wedge \ \mathsf{E}'(\psi) \overset{\mathsf{def}}{=} ((\mathsf{E} \backslash \tau) \cap \psi) \neq \emptyset\}.$$

Let $\Psi_\tau \overset{\mathsf{def}}{=} \{\psi_i \mid i \in \mathcal{I}\}$, and let $n \in \mathbb{N}$ be maximal such that no ψ_i from Ψ_τ is contained in $\underline{\mathfrak{n}}$; let $\psi_1, \dots, \psi_k$ be the only tiles from Ψ_τ that touch $\underline{\mathfrak{n}}$. Then $\mathcal{A} \overset{\mathsf{def}}{=} \{\omega \mid \tau(\omega) \subseteq \underline{\mathfrak{n}}\}$ is the set of runs ω such that, for all $i \in \{1, \dots, k\}$, $\omega \cap \psi_i$ is either empty or contains an event from $\mathsf{E}'(\psi_i)$; clearly, $\mathcal{A} \notin \mathcal{F}_{\underline{\mathfrak{n}}}$, contradicting the assumption $\tau \in \mathfrak{T}$. — For the converse, we have to show that

$$\left.\begin{array}{ccc} \omega_\tau & \subseteq & \omega_{\underline{\mathfrak{n}}} \\ \omega' & \sim_{\underline{\mathfrak{n}}} & \omega \end{array}\right\} \quad \Rightarrow \quad \omega'_\tau \subseteq \omega'_{\underline{\mathfrak{n}}} \tag{20}$$

holds for all $n \in \mathbb{N}$ and $\omega \in \Omega$; but (20) follows since configurations are causally closed. □

The simplest examples of stopping times are *constants*, i.e. prefixes of the form $\underline{\mathfrak{n}}$. A less trivial class, and arguably the most important one, is formed by *hitting times*. For this, we first need some definitions:

Definition 13. *Let $\mathfrak{U}$ be a set of markings for $\mathcal{N}$, and $\mathfrak{C}_{\mathfrak{U}} \subseteq \mathfrak{C}(\overline{\mathsf{N}})$ be the set of the corresponding cuts:*

$$\mathfrak{C}_{\mathfrak{U}} \quad \overset{\mathsf{def}}{=} \quad \left\{\mathbf{c} \in \mathfrak{C} \mid \exists\ \mathsf{M}_{\mathbf{c}} \in \mathfrak{U}: \ \mu_{|\mathbf{c}} = \mathsf{M}_{\mathbf{c}} \circ \pi_{|\mathbf{c}}\right\}.$$

For $\omega \in \Omega(\mathcal{N})$, define the cut $\mathbf{c}_{\mathsf{hit}_{\mathfrak{U}}}$ as $\mathbf{c}_{\mathsf{hit}_{\mathfrak{U}}}(\omega) \overset{\mathsf{def}}{=} \min_{\sqsubseteq}(\mathfrak{C}(\omega) \cap \mathfrak{C}_{\mathfrak{U}})$, if the set of cuts on the right hand side is not empty, and let the random variable $\mathbf{C}_{\mathsf{hit}_{\mathfrak{U}}}$ be the configuration

$$\mathbf{C}_{\mathsf{hit}_{\mathfrak{U}}}(\omega) \quad \overset{\mathsf{def}}{=} \quad \begin{cases} (\mathbf{c}_{\mathsf{hit}_{\mathfrak{U}}}(\omega))^{\Downarrow} & \textit{if } \mathbf{c}_{\mathsf{hit}_{\mathfrak{U}}} \textit{ is defined, and} \\ \omega & \textit{otherwise.} \end{cases} \tag{21}$$

Then the (first) hitting time *for $\mathfrak{U}$ is the prefix*

$$\rho_{\mathfrak{U}} \stackrel{\mathsf{def}}{=} \bigcup_{\omega\in\Omega} \mathbf{C}_{\mathsf{hit}_{\mathfrak{U}}}(\omega)\,.$$

The following lemma ensures that, in our setting, $\mathbf{c}_{\mathsf{hit}_{\mathfrak{U}}}$ and $\rho_{\mathfrak{U}}$ are well-defined; here, an occurrence net $\overline{\mathsf{N}}$ is said to be of *finite width* iff $|\mathbf{c}| \in \mathbb{N}$ for all $\mathbf{c} \in \mathfrak{C}(\overline{\mathsf{N}})$.

Lemma 4. [11, 23] *Let $\overline{\mathsf{N}}$ be an occurrence net of finite width. Then, for all $\omega \in \Omega$, $\mathfrak{C}(\omega)$ is well-ordered by $\sqsubseteq$, and is a conditionally complete lattice.*

We can thus proceed to show:

Theorem 3. *For any set $\mathfrak{U}$ of markings, the hitting time $\tau_{\mathsf{hit}_{\mathfrak{U}}}$ is a stopping time.*

Proof: For all $n \in \mathbb{N}$, one has $\{\omega \mid \omega_{\tau_{\mathsf{hit}_{\mathfrak{U}}}} \subseteq \underline{n}\} = \{\omega \mid \mathbf{C}_{\mathsf{hit}_{\mathfrak{U}}}(\omega) \sqsubseteq \underline{n}\} \in \mathcal{F}_{\underline{n}}$. □

We close our investigation of properties of $\mathfrak{T}$ with the following:

Theorem 4. *$\mathfrak{T}$ is a complete sub-lattice (see Lemma 4) of* Pref.

Proof: Follows directly from Theorem 2 since the tile-respecting property is preserved by arbitrary unions and intersections. □

5.2 Markov Property

The *Markov Property* for stochastic processes means, informally, that the future behavior may depend on the current state, but not on past behavior, i.e. not on any aspect of how the current state has been reached; if any *stopping time* can be taken as the present instant with this property, the process is said to satisfy a *strong Markov Property.* Since the behavior of a Petri Net is intuitively memoryless, it is an important test for our model whether it preserves that property. We will show here how the Markov property carries over to the filtration here, and is satisfied by our model, no matter which cluster measures are chosen. This question is of interest in two respects. Firstly, it shows that the probabilistic cluster unfolding respects the absence of memory intrinsic in Petri nets: remember that the firing rule and, consequently, the entire behavior of a Petri Net is determined by its current marking, independently of the way taken to arrive in that marking. A Petri net $\mathcal{N}$ that started in M_0, reached M and *continues* from marking M afterwards, is indiscernible from the net $\mathcal{N}'$ obtained by replacing M_0 by M; its evolution can be *re-started* from M without loss of information. Now, re-starting a *stochastic* process without loss of information it exactly what is expressed by the Markov property. Secondly, the class of Markov Processes is among those classes of stochastic processes whose asymptotic behavior can best be analyzed.

To reason about the future evolution, we need another family of σ-algebras that abstracts in an appropriate way from the "pre-history" of the process.

Definition 14. *Fix $\tau \in \mathfrak{T}$. For $\mathcal{E} \subseteq (\mathsf{B} \cup \mathsf{E})$ and $\omega, \omega' \in \Omega$, let $\omega \sim_{\mathcal{E}} \omega'$ iff $\omega \cap \mathcal{E}$ and $\omega' \cap \mathcal{E}$ are isomorphic as labeled graphs (i.e. the isomorphism commutes with π, μ and β); note the weakening of "$\sim$" compared to (7). We obtain a σ-algebra $\mathcal{F}_{\mathcal{E}}$ on Ω by letting $\mathcal{X} \in \mathcal{F}_{\mathcal{E}}$ iff:*

$$\left.\begin{array}{r}\omega \in \mathcal{A}\\ \omega' \sim_{\mathcal{E}} \omega\end{array}\right\} \Rightarrow \omega' \in \mathcal{A};$$

so the construction is analogous to (8) but based on a weaker equivalence.

Note that, if $\mathcal{E}$ is a prefix, both definitions for $\mathcal{F}_{\mathcal{E}}$ are equivalent. Let the *future* of τ be the subnet $\mathfrak{Z}_\tau$ of $\overline{\mathsf{N}}$ spanned by $\mathcal{OH}([\mathsf{B} \cup \mathsf{E}] \backslash \tau)$, and the σ-algebra $\mathcal{F}_{\mathfrak{Z}_\tau}$ is defined according to Definition 14. Let the *slice* of τ be the set

$$\mathsf{S}_\tau \stackrel{\mathsf{def}}{=} \bigcup_{\psi_* \in \Psi_*(\tau)} \psi_*,$$

where $\Psi_*(\tau) \overset{\mathsf{def}}{=} \{\psi_* \in \Psi_*\mathbf{in}(\psi) \subseteq \tau \wedge \psi \not\subseteq \tau\}$ is the set of *complete* tiles at the front of τ. For notational convenience, write $\theta_1\tau \overset{\mathsf{def}}{=} \tau \cup \mathsf{S}_\tau$, and define inductively $\theta_{n+1}\tau \overset{\mathsf{def}}{=} \theta\theta_n\tau$ for $n \in \mathbb{N}$. By virtue of Theorem 2, $\{\theta_n\tau \mid n \in \mathbb{N}\} \subseteq \mathfrak{T}$.

Theorem 5. (Strong Markov Property) *For all* $\tau \in \mathfrak{T}$ *and* $\mathcal{A} \in \mathcal{F}_{\mathfrak{Z}_\tau}$*,*

$$\mathbb{P}(\mathcal{A} \mid \mathcal{F}_\tau) \quad = \quad \mathbb{P}(\mathcal{A} \mid \mathsf{M}_\tau)\,. \tag{22}$$

Proof: For $n \in \mathbb{N}_0$, set $\mathsf{S}_n \overset{\mathsf{def}}{=} \mathsf{S}_{\theta_n\tau}$. $\mathfrak{Z}_\tau$ can be represented as

$$\mathfrak{Z}_\tau \quad = \quad \mathcal{OH}(\bigcup_{n\in\mathbb{N}_0} \mathsf{S}_n); \tag{23}$$

$\mathfrak{Z}_\tau$ can in fact be seen as the merger of all unfoldings obtained from the different markings $\mathsf{M}_\tau(\omega)$, where ω varies over Ω. The $\mathcal{OH}(\cdot)$ operator is necessary here, as below, since slices are not condition-bordered. Define

$$\mathfrak{Z}_\tau^{(K)} \overset{\mathsf{def}}{=} \mathcal{OH}(\bigcup_{n=0}^{K} \mathsf{S}_n) \quad \textit{and} \quad \mathfrak{Z}_\tau^{(K+)} \overset{\mathsf{def}}{=} \mathcal{OH}(\bigcup_{n>K} \mathsf{S}_n),$$

thus $\mathfrak{Z}_\tau = \mathfrak{Z}_\tau^{(K)} \cup \mathfrak{Z}_\tau^{(K+)}$. By (23), there exists $K \in \mathbb{N}$ such that $\mathcal{A} \in \mathcal{F}_{\mathfrak{Z}_\tau^{(K)}}$. Hence we will be able to prove (22) by induction over K once we will have established that, for $\omega, \omega' \in \Omega$,

$$\begin{pmatrix} \omega & \sim_{\mathsf{S}_K} & \omega' \\ \wedge\ \omega & \sim_{\mathfrak{Z}_\tau^{(K+)}} & \omega' \end{pmatrix} \quad \Rightarrow \quad \omega \sim_{\mathfrak{Z}_\tau^{(K)}} \omega'. \tag{24}$$

Suppose $\omega \sim_{\mathsf{S}_K} \omega'$ and $\omega \sim_{\mathfrak{Z}_\tau^{(K+)}} \omega'$. Then there is an isomorphism ϕ of labeled graphs *(LGI)* from ω_{S_K} to ω'_{S_K}. By the definition of the cluster process, this implies $\omega \sim_{\mathcal{OH}(\mathsf{S}_K)} \omega'$, and that there is an LGI ϕ^* from $\mathcal{OH}(\omega_{\mathsf{S}_K})$ to $\mathcal{OH}(\omega'_{\mathsf{S}_K})$ that extends ϕ. Since $\omega \sim_{\mathfrak{Z}_\tau^{(K+)}} \omega'$, one thus obtains inductively a unique LGI ϕ_* from $\omega_{\mathfrak{Z}_\tau^{(K+)}}$ to $\omega'_{\mathfrak{Z}_\tau^{(K+)}}$ such that ϕ_* extends ϕ; hence (24) follows.

For $K = 1$, we have $\mathcal{A} \in \mathcal{F}_{\mathsf{S}_1}$. By definition of $\mathcal{F}_{\mathsf{S}_1}$, it suffices to consider $\mathcal{A}$ of the following form: there exists $\omega_0 \in \Omega$ such that

$$\mathcal{A} \quad = \quad \{\omega \mid \omega \sim_{\mathsf{S}_1} \omega_0\}\,. \tag{25}$$

Note that $\sim_{\mathsf{S}_1}$ determines $\omega \cap \mathsf{S}_1$ only up to an isomorphism of labeled graphs; however, the set

$$\mathbf{V}_{\mathcal{A}} \quad \overset{\mathsf{def}}{=} \quad \{\overline{\mathsf{N}}\,[\omega \cap \mathsf{S}_1] \mid \omega \in \mathcal{A}\}$$

of *different* subnets spanned by intersections of runs with S_1 is finite. For each $\mathbf{v} \in \mathbf{V}_{\mathcal{A}}$, choose some $\omega_{\mathbf{v}} \in \Omega$ such that $\mathbf{v}$ is $\overline{\mathsf{N}}[\omega_{\mathbf{v}} \cap \mathsf{S}_1]$. For a tile ψ and a run ω with $\psi \cap \omega \neq \emptyset$, let $\mathsf{e}_\psi(\omega)$ be the unique event of $\psi \cap \omega \neq \emptyset$. With these preparations, define a random variable $\mathbf{X}$ by

$$\mathbf{X}(\omega) \quad \overset{\mathsf{def}}{=} \quad \sum_{\mathbf{v}\in\mathbf{V}_{\mathcal{A}}} \left[\prod_{\psi\in\Psi_{\mathbf{v}}} \mathbb{P}^{\psi}\left(\mathsf{e}_\psi(\omega_{\mathbf{v}})\right) \right], \tag{26}$$

where for $\mathbf{v} \in \mathbf{V}_{\mathcal{A}}$, $\Psi_{\mathbf{v}} \overset{\mathsf{def}}{=} \{\psi \in \Psi \mid \mathbf{in}(\psi) \subseteq \mathbf{c}_\tau(\omega_{\mathbf{v}}) \wedge \omega_{\mathbf{v}} \cap \psi \neq \emptyset\}$; $\mathbb{P}^\psi$ is the tile measure for ψ, and we use the convention that $\mathbb{P}^\psi(\mathsf{e}_\psi(\omega_{\mathbf{v}})) = 0$ whenever $\psi \cap \omega = \emptyset$; this entails that the only non-vanishing summands in (26) are those representing a probabilistic choice in favor of ω. Now, $\mathbf{X}$ is $\mathcal{F}_{\mathsf{S}_1}$-measurable by construction; to show (22), it thus suffices to show

$$\mathbf{X} \quad = \quad \mathbb{P}(\mathcal{A} \mid \mathcal{F}_{\theta\tau})\,. \tag{27}$$

In other words, we have to show that for any $\mathcal{A}' \in \mathcal{F}_\tau$,

$$\mathbf{E}\,[\,\mathbb{1}_{\mathcal{A}'} \cdot \mathbf{X}] \;=\; \mathbb{P}\,(\mathcal{A} \cap \mathcal{A}')\,, \tag{28}$$

where $\mathbf{E}[\,\mathbb{1}_{\mathcal{A}'} \cdot \mathbf{X}]$ is the expectation under $\mathbb{P}$. Again, it suffices to consider the case where $\mathcal{A}'$ is of the form $\mathcal{A}' = \{\omega \mid \omega \sim_{\mathsf{S}_1} \omega_0'\}$ for some fixed $\omega_0' \in \Omega$. By virtue of (25),

$$\int_{\mathcal{A}'} \mathbf{X}\; d\mathbb{P} \;=\; \mathbb{P}\,(\mathcal{A}_\tau) \cdot \sum_{\mathbf{v} \in \mathbf{V}_{\mathcal{A} \cap \mathcal{A}'}} \left[\prod_{\psi \in \Psi_{\mathbf{v}}} \mathbb{P}^{\psi}\,(\mathsf{e}_\psi(\omega_{\mathbf{v}})) \right] = \mathbb{P}\,(\mathcal{A} \cap \mathcal{A}')\,,$$

so we are done; the induction $K \to K+1$ follows analogously. □

Remark 2. *Note that a strong Markov property was also shown in [8], for branching process semantics; the cluster semantics makes the result here more general and simplifies the technical parts of the proof. Theorem 5 states a loss of memory of the evolution that led to the* marking M_τ*; that is, the present state sufficient to determine the law of the future is* M_τ*. The strong Markov Property in [8] is based on a different notion of* "present"*, requiring only partial knowledge of the global state. This is inextricably connected to the branching process semantics used there; for cluster semantics,* M_τ *is the adequate choice of* "present state at τ"*. Not further that, since the constants* $\underline{\mathrm{n}}$ *are stopping times, Theorem 5 implies also a* weak Markov property.

5.3 Causal Independence

Although the concurrency relation co is often referred to as *independence*, two concurrent events need not be stochastically independent. To see this, consider the events A and D in Parts (C) and (D) of Figure 7: $\{A, D\}$ is a co-set, yet the occurrence of A entails that of D, i.e. $(A \in \omega) \Rightarrow (D \in \omega)$; and thus $\mathbb{P}(A \cap D) = \mathbb{P}(A)$, making stochastic independence impossible except for the trivial cases $\mathbb{P}(D) = 1$ and $\mathbb{P}(D) = \mathbb{P}(A) = 0$. Note: This and the example of the introduction cover two main types of *confusion* in the sense of [38]

The highest possible degree of independence in a general Petri net is that of jointly enabled steps belonging to independent clusters $\gamma_1, \ldots, \gamma_k$; this is illustrated in its purest form by the initial example on the left hand side of Figure 1. In general, connected nets, this independence is necessarily *conditional*:

→ given the current marking M and

→ given the choice, in M, of an $\mathbb{I}$-clique Δ containing $\gamma_1, \ldots, \gamma_k$ by the scheduling cluster $\mathcal{N}^{\mathrm{T}}$.

For the latter point, the choices of $\gamma_1, \ldots, \gamma_k$ are also pairwise conditionally independent given the knowledge of whether or not their respective immediate neighbors are selected: this is the Markov field requirement (10), which indicates the highest degree of independence possible within one cluster. Moving on, Markov fields on all clusters (including $\mathcal{N}^{\mathrm{T}}$) give the highest degree of independence on *all* levels; any modification of, e.g., the firing rule that destroys a Markov Field thus makes the evolution of the Petri net less *local*, by introducing new non-local dependencies.

6 Conclusions

The results presented here lay the foundations for a probabilistic analysis of concurrent systems in logical time. To develop a valid probabilistic model of concurrent runs, it was essential to change the semantics used in describing Petri net behavior to *cluster semantics*. We have explored the "semi-local" cluster view from [27] here, in conjunction with and motivated by probabilistic considerations. The essential property that makes cluster unfoldings useful as a carrier of the probabilistic model is the decomposition into tiles, which allows structural insight and easier manipulation.

Randomizing the choice of the cluster step by Gibbs potentials allows to preserve independence, see the above discussion.In fact, it is interesting to note that the Gibbs potential approach – had we applied it without previously thinking of clusters – leads naturally to the partition of a net into clusters, since all conflict cliques are contained in a cluster; hence the cluster unfoldings can be seen as a *"Gibbs semantics"* for Petri nets.

The role of the cluster net $\mathcal{N}^{\mathrm{T}}$ is to give some structure to the action of the clusters on the state of the net. It can do so in more or less equitable ways. One may say that $\mathcal{N}^{\mathrm{T}}$ distributes active time over the "players", possibly forcing one or the other to let time pass (and see tokens arrive) without a possibility to progress in its turn; this behavior, which leads to structurally unbalanced processes, is excluded if the cluster net decisions are non-degenerate and i.i.d.

Acknowledgments: I wish to thank A. Benveniste and E. Fabre for substantial comments, discussions, and encouragement.

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] A. Aghasaryan, E. Fabre, A. Benveniste, R. Boubour, and C. Jard. A Petri net approach to fault detection and diagnosis in distributed systems.*Proceedings CDC* 1997.

[4] A. Aghasaryan, E. Fabre, A. Benveniste, R. Boubour, and C. Jard. Fault detection and diagnosis in distributed systems: an approach by partially stochastic petri nets. *Discrete event dynamic systems* **8**:203–231, 1998.

[5] L. de Alfaro. From Fairness to Chance. *Elec. Notes on Theor. Comp. Sci.* **22**, Elsevier, 2000.

[6] L. de Alfaro. Stochastic Transition Systems. *CONCUR 98*, LNCS **1466**:423-438, Springer, 1998.

[7] M.S. Andersland and D. Teneketzis. Information Structures, Causality, and Nonsequential Stochastic Control. *SIAM J. of Contr. and Opt.* Part I: Design-Independent Properties. **30**(6):1447–1475, 1992. Part II: Design-Dependent Properties. **32**(6):1726–1751, 1994.

[8] A. Benveniste, S. Haar, and E. Fabre. Markov Nets: probabilistic Models for Distributed and Concurrent Systems. *Rapport de Recherche* INRIA **4235**, September 2001.

[9] A. Benveniste, E. Fabre, S. Haar, and C. Jard. Diagnosis of Asynchronous Discrete Event Systems, a Net Unfolding Approach. *Rapport de Recherche* INRIA **4181**, July 2001.

[10] A. Benveniste, B.C. Levy, E. Fabre, and P. LeGuernic. A calculus of stochastic systems: specification, simulation, and hidden state estimation. *Theoretical Computer Science* **152**:171–217, 1995.

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

[12] K. L. Chung. *A Course in Probability Theory.* Academic Press, 1974.

[13] C. Derman. *Finite State Markovian Decision Processes.* Academic Press, 1970.

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

[15] R. Dobrushin. The description of a random field by means of conditional probabilities and conditions of its regularity. Theory Prob. Appl. **13**:197–224, 1968.

[16] S. Dolev, A. Israeli, and S. Moran. Analyzing expected time by scheduler-luck games. *IEEE Transactions on Parallel and Distributed Systems*, **8**(4):424–440, April 1997.

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

[18] G. Rozenberg and J. Engelfriet. Elementary Net Systems. *Lectures on Petri Nets I: Basic Models.* LNCS **1491**:12–121, Springer, 1998.

[19] J. Esparza and S. Römer. An unfolding algorithm for synchronous products of transition systems. *CONCUR'99*, LNCS **1664**. Springer, 1999.

[20] J. Esparza, S. Römer, and W. Vogler. An Improvement of McMillan's Unfolding Algorithm. *TACAS '96*, LNCS **1055**:87–106, Springer 1996.

[21] J. Esparza, S. Römer, and W. Vogler. An Improvement of McMillan's Unfolding Algorithm. Extended version of [20]; to appear in *Formal Methods in System Design.*

[22] J. Esparza. Model Checking Using Net Unfoldings. *Sci. of Comp. Prog.* **23**:151–195, 1994.

[23] C. Fernández and P. S. Thiagarajan. A Lattice Theoretic View of K-Density. G. Rozenberg (ed.), *Advances in Petri Nets 1984.* LNCS **188**:139–153, Springer, Berlin, 1985.

[24] H.-O. Georgii. *Gibbs Measures and Phase Transitions.* de Gruyter, Berlin/New York 1988.

[25] S. Haar. Branching Processes of general S/T-Systems. *Elec. Notes in Theor. Comp. Sci.* **18**, 1998.

[26] S. Haar. Occurrence Net Logics. *Fundamenta Informaticae* **43**:105–127, 2000.

[27] S. Haar. Clusters, confusion and unfoldings. *Fundamenta Informaticae* **47**(3-4):259–270, 2001.

[28] C.N. Hadjicostis and G.C. Verghese. Monitoring discrete event systems using Petri Net embeddings. *Proc. ICATON 99*, LNCS **1639**, 188–208.

[29] K. Heljanko, V. Khomenko and M. Koutny. Parallelisation of the Petri Net Unfolding Algorithm. To appear in *TACAS* 2002.

[30] K. McMillan. Using Unfoldings to avoid the state explosion problem in the verification of asynchronous circuits. *4th Workshop on Computer Aided Verification*, pp. 164–174, 1992.

[31] R. Kindermann and J. L. Snell. *Markov Random Fields and their Applications.* American Mathematical Society, Contemporary Mathematics Vol. 1, Providence (RI), 1980.

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

[33] M.O. Rabin. Probabilistic Automata. *Information and Control* **6**:230–245, 1963.

[34] W. Reisig. *Petri Nets*, vol. 4 of *EATCS Monographs on Theor. Comp. Sci.* Springer, 1985.

[35] R. Segala. Verification of Randomized Distributed Algorithms, Lectures on Formal Methods and Performance Analysis. LNCS **2090**:232-260, Springer 2001.

[36] E. Bandini and R. Segala. Axiomatizations for Probabilistic Bisimulation. *Proc. 28th ICALP*, LNCS **2076**:370-381, Springer 2001.

[37] R. Segala. A Compositional Trace-Based Semantics for Probabilistic Automata. *CONCUR'01*, LNCS **962**:234-248, Springer 1995.

[38] E. Smith. On the border of causality: Contact and Confusion. *TCS* **153**:245–270, 1996.

[39] G. Stremersch. *Supervision of Petri Nets*, Kluwer, Boston/Dordrecht/London, 2001.

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

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

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