



HAL
open science

Branching cells as local states for event structures and nets: probabilistic applications

Samy Abbes, Albert Benveniste

► **To cite this version:**

Samy Abbes, Albert Benveniste. Branching cells as local states for event structures and nets: probabilistic applications. Foundations of Software Science and Computation Structures, Apr 2005, Edinburgh, United Kingdom. pp.95-109. hal-00350930

HAL Id: hal-00350930

<https://hal.science/hal-00350930v1>

Submitted on 7 Jan 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Branching Cells as Local States for Event Structures and Nets: Probabilistic Applications

Samy Abbes* and Albert Benveniste**

IRISA Campus de Beaulieu,
35042 Rennes Cedex. France

Abstract. We study the concept of choice for true concurrency models such as prime event structures and safe Petri nets. We propose a dynamic variation of the notion of cluster previously introduced for nets. This new object is defined for event structures, it is called a *branching cell*. Our aim is to bring an interpretation of branching cells as a right notion of “local state”, for concurrent systems.

We illustrate the above claim through applications to probabilistic concurrent models. In this respect, our results extends in part previous work by Varacca-Völzer-Winskel on probabilistic confusion free event structures. We propose a construction for probabilities over so-called *locally finite* event structures that makes concurrent processes probabilistically independent—simply attach a dice to each branching cell; dices attached to *concurrent* branching cells are thrown independently. Furthermore, we provide a true concurrency generalization of Markov chains, called *Markov nets*. Unlike in existing variants of stochastic Petri nets, our approach randomizes Mazurkiewicz traces, not firing sequences. We show in this context the Law of Large Numbers (LLN), which confirms that branching cells deserve the status of local state.

Our study was motivated by the stochastic modeling of fault propagation and alarm correlation in telecommunications networks and services. It provides the foundations for probabilistic diagnosis, as well as the statistical distributed learning of such models.

1 Introduction

The study we present in this paper was motivated by algorithmic problems of distributed nature encountered in the area of telecommunications network and service management [4], in particular distributed alarm correlation and fault diagnosis. This problem consists in reconstructing the hidden history of the distributed system from partial observations (the alarms). The supervision architecture is distributed and comprises several supervisors acting as peers and communicating asynchronously.

* ISR, A. V. Williams Building, University of Maryland, College Park, MD 20742, USA; work performed while this author was with IRISA/Université de Rennes 1.

** IRISA/INRIA, benveniste@irisa.fr, <http://www.irisa.fr/sigma2/benveniste/>

True concurrency is essential in these algorithms: interleaving semantics is not adequate for such large distributed systems. States need to be local. Time is totally ordered at each network node, but only partially ordered by causality between nodes. Due to unavoidable ambiguity in diagnosis, nondeterminism is solved by seeking for the “most likely” solutions of the diagnosis problem. This requires having a probabilistic setting at hand.

While searching for existing models in the literature, we found very few approaches meeting our requirements. *Stochastic Petri nets* [6] and their variants are useful for performance evaluation. This model typically randomizes the holding time in places or the firing time at transitions. Making reference to a global time causes some probabilistic coupling to occur between subsystems that otherwise do not interact. *Probabilistic process algebras* [7] or *probabilistic automata* [11] are related to so-called Markov Decision Processes from applied probability theory, they rely on interleaving semantics and do not meet our needs either. In those models, interactions occur via synchronized actions and are subject to nondeterminism. In contrast, probabilistic choices are purely private, occur between interactions and do not conflict with these. Whereas this is perfectly adequate, e.g, for testing or security protocols [8, 9], this is not convenient for modeling the uncertain occurrence and propagation of faults and alarms in telecommunications networks.

Concurrent probabilistic models is a recent area of research meeting our requirements. Runs of concurrent systems are randomized without reference to a global clock, and with a true-concurrent semantics. Fundamental difficulties have lead to restrict to models with limited concurrency, e.g., *confusion free event structures* [14, 13]. *Distributed probabilistic event structures* and *Markov nets* are studied in [1], following an approach initiated in [3]; these approaches address event structures with confusion.

It appears that the very key for the analysis of probabilistic choice in true-concurrent models are the informal concepts of “concurrent local state” and “concurrent local choices”. In this paper, we investigate these notions for safe Petri nets and prime event structures. We show that so-called *branching cells* introduced in [1] for event structures provide the answer. Informally, for an event structure, branching cells are minimal subsets of events closed under immediate conflict. Processes are *dynamically* decomposed by branching cells: in different executions, the same event can belong to different branching cells. Branching cells differ from clusters [5], which are statically defined on nets.

We apply the notion of branching cell to the definition and construction of concurrent probabilistic models. The probabilities we construct in this way satisfy the following essential requirement regarding concurrency: *parallel local processes are made independent in the probabilistic sense, conditionally on their common past*. Such probabilities deserve the name of *distributed probabilities*. They generalize to event structures with confusion the notion of *valuation with independence* from [13]. When applied to event structure obtained by unfolding safe Petri nets, this yields *Markov nets*, a probabilistic form of Petri nets com-

pliant with true concurrency. We prove a Markov property and a Law of Large Numbers for Markov nets, in which branching cells play the role of local states.

The paper is organized as follows. Branching cells for prime event structures are introduced in Section 2, together with their properties. Their use for the definition and construction of concurrent probabilistic models is demonstrated in Section 3. In Section 4, Markov nets are introduced in order to state the Markov property and the Law of Large Numbers.

2 Branching Cells and Their Properties

A prime event structure [10] is a triple $\mathcal{E} = (E, \preceq, \#)$ satisfying the following properties. (E, \preceq) is a partial order. The elements of E are called *events* and E is at most countable. $\#$ is the *conflict* relation on E ; it is a binary relation that is symmetric and irreflexive, and satisfies the following axiom: $\forall x, y, z \in E, x\#y$ and $y \preceq z$ together imply $x\#z$. A subset $A \subseteq E$ is said to be a *prefix* if it is downwards closed: $\forall x \in E, \forall y \in A, x \preceq y \Rightarrow x \in A$. Finally, a prefix v is called a *configuration* of \mathcal{E} if it is *conflict-free*, i.e., if $\# \cap (v \times v) = \emptyset$. Configurations are partially ordered by inclusion, and we denote by $\mathcal{V}_{\mathcal{E}}$ the poset of the *finite* configurations of \mathcal{E} . We denote by $\Omega_{\mathcal{E}}$ the set of maximal configurations of \mathcal{E} —this set is nonempty, due to Zorn’s Lemma. A subset $F \subseteq E$ implicitly defines a sub-event structure $(F, \preceq_F, \#_F)$ of \mathcal{E} with causality and conflict relations inherited by:

$$\preceq_F = \preceq \cap (F \times F), \quad \#_F = \# \cap (F \times F),$$

and we shall freely write $F, \mathcal{V}_F,$ and Ω_F to denote this event structure and its set of finite and maximal configurations, respectively. For $e \in E, [e] \triangleq \{e' \in \mathcal{E} : e' \preceq e\}$ denotes the smallest configuration containing e . For v a finite or infinite configuration of \mathcal{E} , we set $E^v \triangleq \{e \in E \setminus v : \forall e' \in v, \neg(e\#e')\}$. We denote by \mathcal{E}^v the induced event structure and we call it the *future* of v . Throughout the paper, we assume that \mathcal{E} satisfies the following assumption:

Assumption 1. *Configuration $[e]$ is finite for every event e . For every $v \in \mathcal{V}_{\mathcal{E}}, \text{Min}_{\preceq}(E^v)$ contains finitely many events.*

The first part of Assumption 1 is very standard, it says that every event has finitely many causal predecessors. The second part of the assumption expresses that any finite configuration enables only finitely many events. The *concurrency* relation on E , denoted by \parallel , is defined as the reflexive closure of $(\mathcal{E} \times \mathcal{E}) \setminus (\# \cup \preceq \cup \succeq)$.

A central concept in defining probabilities is the notion of choice. Choice is therefore a key concept in this paper; it is captured by the notion of immediate conflict we recall next. The *immediate conflict* relation $\#_{\mu}$ on E is defined by:

$$\forall e, e' \in E, \quad e\#_{\mu}e' \quad \text{iff} \quad ([e] \times [e']) \cap \# = \{(e, e')\}. \quad (1)$$

Definition 1 (stopping prefix). *A prefix B of \mathcal{E} is called a stopping prefix iff it is closed under immediate conflict.*

\mathcal{E} is called *locally finite* iff for each event e of \mathcal{E} , there exists a finite stopping prefix B containing e . The following condition is assumed throughout this paper:

Assumption 2. \mathcal{E} is locally finite.

Locally finite event structures have not been considered by authors so far. We shall see at the end of this section that confusion freeness implies local finiteness.

Stopping prefixes B satisfy the following property (see [1–Ch.3, l-3.1]):

$$\Omega_B = \{\omega \cap B \mid \omega \in \Omega_{\mathcal{E}}\}. \quad (2)$$

Although the inclusion \subseteq always holds, not every prefix does satisfy the equality of property (2). Take for instance $E = \{a, b\}$ with $a \# b$. Consider prefix $P = \{a\}$ and maximal configuration $\omega = \{b\}$. Then $\omega \cap P = \emptyset$ is not maximal in P .

Clearly, the set of all stopping prefixes is a complete lattice. However, stopping prefixes are not stable under concatenation: if B is a stopping prefix of \mathcal{E} , $v \in \Omega_B$, and B^v is a stopping prefix of \mathcal{E}^v , then $B \cup B^v$ is generally not a stopping prefix of \mathcal{E} . As a consequence, the concatenation of v and of a configuration stopped in \mathcal{E}^v is not stopped in \mathcal{E} in general. Roughly speaking, the class of stopped configurations is not closed under concatenation, which is inconvenient. The notions of recursively stopped configuration and branching cell we introduce next overcome this drawback.

Definition 2 (stopped and recursively stopped configurations).

1. A configuration v of \mathcal{E} is said to be *stopped* if there is a stopping prefix B such that $v \in \Omega_B$.
2. Call *recursively stopped* a configuration v of \mathcal{E} such that there exists a finite nondecreasing sequence $(v_n)_{0 \leq n \leq N}$ of configurations, where $v_0 = \emptyset$, $v_N = v$, and for $n < N$, $v_{n+1} \setminus v_n$ is a finite stopped configuration of the future \mathcal{E}^{v_n} of v_n . The set of all finite recursively stopped configurations is denoted by $\mathcal{W}_{\mathcal{E}}$, or simply \mathcal{W} if no confusion can occur.

The class of recursively stopped configurations is the smallest class of configurations that contains stopped configurations and is closed under concatenation (see the examples at the end of this section).

Definition 3 (branching cell). Stopping prefix B is called *initial* iff \emptyset is the only stopping prefix strictly contained in B . Call *branching cell* of \mathcal{E} any initial stopping prefix of \mathcal{E}^v , where v ranges over \mathcal{W} . The set of all branching cells of \mathcal{E} is denoted by $X_{\mathcal{E}}$ (or simply X when no confusion can occur). Branching cells are generically denoted by the symbol x .

Informally, branching cells are minimal subsets of events closed under immediate conflict. For $v \in \mathcal{W}$, denote by $\delta(v)$ the set of branching cells that are initial prefixes of \mathcal{E}^v . Clearly, branching cells of $\delta(v)$ do not overlap (in general, branching cells may overlap, see the examples at the end of this section). Consider the following map Δ , called the *covering* map of \mathcal{E} :

$$\begin{aligned} \text{for } v \in \mathcal{W}: \quad \Delta(v) &\triangleq \overline{\Delta}(v) \setminus \delta(v), \\ \text{where } \overline{\Delta}(v) &\triangleq \{x \in \delta(v') \mid v' \in \mathcal{W}, v' \subseteq v\}. \end{aligned} \quad (3)$$

We list some properties of branching cells. The proof of Th. 4 is given in the Appendix, the remaining proofs are found in the extended version [2].

Theorem 1. *If B is a stopping prefix of \mathcal{E} , then $X_B \subseteq X_{\mathcal{E}}$ and $\mathcal{W}_B \subseteq \mathcal{W}_{\mathcal{E}}$. Furthermore, the covering maps Δ and Δ_B respectively defined on \mathcal{W} and \mathcal{W}_B coincide on \mathcal{W}_B .*

Theorem 2. *For every $v \in \mathcal{W}$, $X_{\mathcal{E}^v} \subseteq X_{\mathcal{E}}$. For $v \subseteq v'$ two finite recursively stopped configurations, $v' \setminus v$ is recursively stopped in \mathcal{E}^v . Denote by Δ^v the covering map (3) defined on \mathcal{E}^v . We have:*

$$\Delta(v') = \Delta(v) \cup \Delta^v(v' \setminus v), \quad \text{and} \quad \Delta(v) \cap \Delta^v(v' \setminus v) = \emptyset. \quad (4)$$

Theorem 3. *Branching cells recursively cover stopped configurations, i.e.:*

$$\forall v \in \mathcal{W}, \quad v = \bigcup_{x \in \Delta(v)} v \cap x, \quad (5)$$

and, for each $x \in \Delta(v)$, $v \cap x$ is an element of Ω_x .

Theorem 4. *Let ξ be a subset of $\delta(\emptyset_{\mathcal{E}})$, where $\emptyset_{\mathcal{E}}$ denotes the empty configuration of \mathcal{E} . The formula*

$$B_{\xi} \triangleq \bigcup_{x \in \xi} x \quad (6)$$

defines a stopping prefix of \mathcal{E} , whose set of finite configurations $\mathcal{V}_{B_{\xi}}$ and maximal configurations $\Omega_{B_{\xi}}$ respectively decompose as:

$$\mathcal{V}_{B_{\xi}} = \prod_{x \in \xi} \mathcal{V}_x \quad \text{and} \quad \Omega_{B_{\xi}} = \prod_{x \in \xi} \Omega_x. \quad (7)$$

Call this a prefix of \mathcal{E} of the form (6), where $\xi \subseteq \delta(\emptyset_{\mathcal{E}})$. The complete lattice of thin prefixes has finite upper bound.

Comments. Theorem 1 expresses that recursively stopped configurations and branching cells are stable under restriction to stopping prefixes.

Theorem 2 expresses that recursively stopped configurations and branching cells are stable under restriction to the futures \mathcal{E}^v of elements $v \in \mathcal{W}$. Equation (4) says that covering maps are incremental with respect to the future.

Theorem 3 is self explanatory. Remark that the property $v \cap x \in \Omega_x$ extends the property $\omega \cap B \in \Omega_B$ stated by Eqn. (2).

The product forms given in Th. 4 show that branching cells are traversed by local processes that are both *concurrent* and *independent*: in the future of v , local decisions taken in a branching cell $x \in \delta(v)$ do not influence the range of possible local decisions that can be taken in other branching cells of $\delta(v)$. In other words, choices in different concurrent branching cells are made by independent and non-communicating agents. Section 3 adds a probabilistic interpretation to this.

Theorem 4 is stated only for thin prefixes that “begin” the event structure. However, Th. 4 can be recursively applied in the futures \mathcal{E}^v , for $v \in \mathcal{W}$, with $\delta(v)$ playing the role of $\delta(\emptyset_{\mathcal{E}})$.

Finally, the finiteness of the above introduced objects follows from our assumptions: the finiteness of branching cells follows from Assumption 2, and the finiteness of the upper bound $\bigcup_{\xi} B_{\xi}$ of thin prefixes follows from Assumption 1 (see the proof of Th. 4 in the Appendix).

Examples. For all examples of this paper, we write (abc) to denote the configuration $\{a, b, c\}$.

The event structure \mathcal{E} shown in Figure 1–left has two nonempty stopping prefixes: $\{a, b\}$ and $\{a, b, c, d, e\}$. Its stopped configurations are \emptyset , (a) , (b) , (a, c, e) , (b, d) , and (b, c, e) . Let us determine the recursively stopped configurations and the branching cells of \mathcal{E} . Since \mathcal{E} has a unique initial stopping prefix $\delta(\emptyset) = \{\{a, b\}\}$, it follows that (a) and (b) are recursively stopped. The future $\mathcal{E}^{(a)}$ is the event structure $\{c, e\}$ with empty conflict and causality; it has two initial stopping prefixes: $\delta(a) = \{\{c\}, \{e\}\}$. Therefore (ac) and (ae) are recursively stopped, as well as (ace) . The future of (ace) is empty. The future $\mathcal{E}^{(b)}$ is given by: $\mathcal{E}^{(b)} = c \rightsquigarrow d \rightsquigarrow e$, with a unique initial stopping prefix: $\delta(b) = \{\{c, d, e\}\}$. Therefore (bd) and (bce) are also recursively stopped. The futures of (bd) and of (bce) are empty, so we are done: $\mathcal{W} = \{\emptyset, (a), (b), (ac), (ae), (ace), (bd), (bce)\}$. Note that (ac) and (ae) are recursively stopped but *not* stopped. Note also that configurations (bc) and (be) are *not* recursively stopped. Finally, the set of all branching cells is $\{\{a, b\}, \{c\}, \{e\}, \{c, d, e\}\}$.

The event structure depicted in Figure 1–middle illustrates the concurrency of branching cells of $\delta(\emptyset)$. Note that some minimal events belong to no initial branching cell.

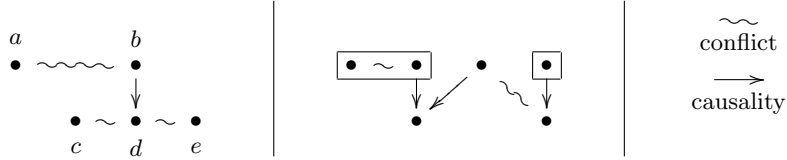


Fig. 1. Left: configuration (ac) is recursively stopped, with associated sequence $(\emptyset, (a), (ac))$ according to Definition 2; however, (ac) is not stopped. Middle: branching cells of $\delta(\emptyset)$ are depicted by frames

Local Finiteness Relaxes Confusion Freeness. Recall that event structure \mathcal{E} is said to be *confusion free* if \mathcal{E} satisfies the Q axiom of concrete domains [10]. Equivalently, \mathcal{E} is confusion free iff [13]:

1. $\#_{\mu}$ is transitive,
2. for all $e, e' \in \mathcal{E}$: $e \#_{\mu} e' \Rightarrow [e] \setminus \{e\} = [e'] \setminus \{e'\}$.

Define, for every event $e \in E$:

$$F(e) = \{f \in E : e \#_{\mu} f\}, \quad B(e) = \bigcup_{f \in [e]} F(f).$$

The second part of Assumption 1 together with point 2 above imply that every set $F(f)$ is finite. It follows that $B(e)$ is finite, and point 1 implies that $B(e)$ is a stopping prefix, that contains e . This holds for every event e , so \mathcal{E} is locally finite. Moreover every finite configuration is stopped, and therefore recursively stopped. The set of branching cells is equal to $\{F(e) : e \in E\}$, which forms a partition of E . Such simple properties fail for event structures with confusion. For example, in the event structure depicted in Figure 1–left, branching cells $\{c\}$ and $\{c, d, e\}$ possess a nonempty intersection. For confusion free event structures, branching cells reduce to the *cells* defined in [13].

To summarize, confusion free event structures are locally finite, but the converse is not true. Locally finite event structures appear as event structures with “finite confusion”.

3 Application to Probabilistic Event Structures

We recall that a *probabilistic event structure* is a pair $(\mathcal{E}, \mathbb{P})$ with \mathbb{P} a probability measure¹ on the space Ω of maximal configurations of \mathcal{E} . We shall prove that a probabilistic event structure can be naturally defined from the new notion of *locally randomized* event structure (Th. 5). The construction performed below adds a probabilistic interpretation to the properties of branching cells and of recursively stopped configurations.

Definition 4 (locally randomized event structure). *A locally randomized event structure is a pair $(\mathcal{E}, (p_x)_{x \in X})$, where X is the set of branching cells of \mathcal{E} , and for each $x \in X$, p_x is a probability over Ω_x .*

Let $(\mathcal{E}, (p_x)_{x \in X})$ be a locally randomized event structure. For $F \subseteq E$ a sub-event structure of \mathcal{E} , denote by X_F the set of all branching cells of F . Call F *well formed* if it is finite and such that $X_F \subseteq X_{\mathcal{E}}$. Note that finite stopping prefixes are well formed according to Th. 1. For F a well formed, set:

$$\text{for } \omega_F \in \Omega_F : \quad \mathbb{P}_F(\omega_F) = \prod_{x \in \Delta(\omega_F)} p_x(\omega_F \cap x), \quad (8)$$

which is well defined since, according to Th. 3, $\omega_F \cap x \in \Omega_x$.

Lemma 1. *If $B = B_{\xi}$ is a thin prefix (see Th. 4), then \mathbb{P}_B is the direct product of the p_x 's, for x ranging over ξ . In particular, \mathbb{P}_B is a probability.*

¹ The σ -algebra considered is the Borel σ -algebra generated by the Scott topology on Ω , see [1] for details. In the remaining of the paper, we do not mention the σ -algebras considered since they are always canonical.

Proof. This is a direct consequence of Eqn. (8) and Th. 4. \diamond

Lemma 2. *If $F \subseteq E$ is a well formed sub-event structure, then \mathbb{P}_F is a probability. In particular, for each stopping prefix B , \mathbb{P}_B is a probability.*

Proof. We show that \mathbb{P}_F is a probability by induction on integer $n_F = \sup_{\omega_F \in \Omega_F} \text{Card}\Delta(\omega_F) < \infty$. The result is a direct consequence of Lemma 1 for $n_F \leq 1$. Assume it holds until $n \geq 1$, and let F be well formed and such that $n_F \leq n + 1$. Consider the (finite) upper bound D of thin prefixes of F . Applying property (2) to D yields the following decomposition for Ω_F : $\Omega_F = \bigcup_{v \in \Omega_D} \{v\} \times \Omega_{F^v}$. Moreover, for each $v \in \Omega_D$ and $\omega' \in \Omega_{F^v}$, and setting $\omega = v \cup \omega'$, we obtain by Th. 2:

$$\Delta(\omega) = \Delta(v) \cup \Delta^v(\omega'), \quad \Delta(v) \cap \Delta^v(\omega') = \emptyset. \quad (9)$$

Formulas (8) and (9) together imply:

$$\sum_{\omega \in \Omega_F} \mathbb{P}_F(\omega) = \sum_{v \in \Omega_D} \mathbb{P}_D(v) \left(\sum_{\omega' \in \Omega_{F^v}} \mathbb{P}_{F^v}(\omega') \right). \quad (10)$$

It follows from Th. 2 that for each $v \in \Omega_D$, the future F^v of v in F satisfies $X_{F^v} \subseteq X_F \subseteq X_{\mathcal{E}}$. Formula (9) implies that $n_{F^v} \leq n$. Hence we can apply the induction hypothesis to F^v and obtain $\sum_{\omega' \in \Omega_{F^v}} \mathbb{P}_{F^v}(\omega') = 1$. From Lemma 1 we get: $\sum_{v \in \Omega_D} \mathbb{P}_D(v) = 1$. This, together with Eqn. (10), implies $\sum_{\omega \in \Omega_F} \mathbb{P}_F(\omega) = 1$, which completes the induction. \diamond

Corollary 1. *Let $B \subseteq B'$ be two finite stopping prefixes of \mathcal{E} . The following formula holds:*

$$\forall \omega_B \in \Omega_B : \mathbb{P}_B(\omega_B) = \sum_{\omega' \in \Omega_{B'}, \omega' \supseteq \omega_B} \mathbb{P}_{B'}(\omega'). \quad (11)$$

Proof. Let ω_B be an element of Ω_B , and denote by $B'' \triangleq B'^{\omega_B}$ the future of ω_B in B' . Then $\{\omega' \in \Omega_{B'} : \omega' \supseteq \omega_B\}$ is one to one with $\Omega_{B''}$. Eqn. (4) gives $\Delta(\omega') = \Delta(\omega_B) \cup \Delta^{\omega_B}(\omega' \setminus \omega_B)$, whence:

$$\sum_{\omega' \in \Omega_{B'}, \omega' \supseteq \omega_B} \mathbb{P}_{B'}(\omega') = \mathbb{P}_B(\omega_B) \sum_{z \in \Omega_{B''}} \mathbb{P}_{B''}(z). \quad (12)$$

From Lemma 2 applied to finite event structure B'' , the sum on the right hand side of (12) equals 1, which implies (11). \diamond

Theorem 5. *Let $(\mathcal{E}, (p_x)_{x \in X})$ be a locally randomized event structure. Then there exists a unique probabilistic event structure $(\mathcal{E}, \mathbb{P})$ such that, for every finite stopping prefix B :*

$$\forall v \in \Omega_B, \quad \mathbb{P}(\{\omega \in \Omega : \omega \supseteq v\}) = \mathbb{P}_B(v), \quad (13)$$

where \mathbb{P}_B is defined by Eqn. (8).

Proof. Corollary 1 expresses that the family (Ω_B, \mathbb{P}_B) , where B ranges over the set of finite stopping prefixes, is a *projective system* of (finite) probability spaces. It is proved in [1–Ch.2] that, under Assumption 2, this projective system defines a unique probability \mathbb{P} on $\Omega_{\mathcal{E}}$ that extends this projective system, i.e., satisfies Eqn. (13). \diamond

Probabilistic Future and Distributed Probabilities. So far we have shown how to construct probabilistic event structures from locally randomized event structures. Conversely, each probability \mathbb{P} over \mathcal{E} , such that $\mathbb{P}(v) > 0$ for every finite configuration v , defines a family $(p_x)_{x \in X}$ of local probabilities associated to branching cells as follows, for $x \in X$ and $\omega_x \in \Omega_x$:²

$$p_x(\omega_x) \triangleq \frac{\mathbb{P}(\{\omega \in \Omega_{\mathcal{E}} : x \in \overline{\Delta}(\omega), \omega \cap x = \omega_x\})}{\mathbb{P}(\{\omega \in \Omega_{\mathcal{E}} : x \in \overline{\Delta}(\omega)\})}. \quad (14)$$

Of course, the following natural question arises: is it true that the family $(p_x)_{x \in X}$ conversely induces \mathbb{P} through Eqn. (8) and Th. 5? Not in general. The following Th. 6, which proof is found in [1–Ch.4], provides the answer.

For $(\mathcal{E}, \mathbb{P})$ a probabilistic event structure, consider the *likelihood* function q defined on the set of finite configurations by:

$$\forall v \in \mathcal{V}_{\mathcal{E}}, \quad q(v) \triangleq \mathbb{P}(\{\omega \in \Omega_{\mathcal{E}} : \omega \supseteq v\}). \quad (15)$$

For v a finite configuration, the *probabilistic future* $(\mathcal{E}^v, \mathbb{P}^v)$ is defined by

$$\mathbb{P}^v(\cdot) \triangleq \frac{1}{q(v)} \mathbb{P}(\cdot).$$

The associated likelihood q^v is given by $q^v(w) = \frac{1}{q(v)} q(v \cup w)$, for w ranging over the set of finite configurations of \mathcal{E}^v .

Definition 5 (distributed probability). A probability \mathbb{P} is called *distributed* iff, for each recursively stopped configuration v , and each thin prefix B_{ξ}^v in \mathcal{E}^v , the following holds:

$$\forall \omega \in \Omega_{B_{\xi}^v}, \quad q^v(\omega) = \prod_{x \in \xi} p_x(\omega \cap x) \quad (16)$$

where p_x is defined from \mathbb{P} by using (14).

Theorem 6. Let $(\mathcal{E}, \mathbb{P})$ be a probabilistic event structure, and let $(p_x)_{x \in X}$ be defined from \mathbb{P} by using (14). The construction of Th. 5 induces again \mathbb{P} iff \mathbb{P} is a distributed probability. In this case, the likelihood function is given on \mathcal{W} by: $q(v) = \prod_{x \in \Delta(v)} p_x(v \cap x)$.

Remark that the likelihood given in Th. 6 extends the original formula (8). Th. 6 also shows that, for confusion-free event structures, the *valuations with independence* defined in [13] are equivalently defined as likelihoods (15) associated with distributed probabilities.

² The condition $p(v) > 0$ is stated here for simplicity, it can be removed with some more technical effort.

Comment. Eqn. (16), which characterizes distributed probabilities, has the following interpretation. Because of the absence of conflicts, and conditionally on a partial execution $v \in \mathcal{W}$, the local choices inside the different branching cells belonging to $\delta(v)$ are performed independently from one another. Eqn. (16) is the probabilistic counterpart of the concurrency of branching cells, stated by Eqn. (7) in Th. 4.

4 Markov Nets

In this section, we apply the previous results to event structures arising from the unfolding of safe and finite Petri nets. *Markov nets* are introduced and briefly studied. Proofs of the results stated in this section as well as additional results can be found in [1], Chapters 5–7.

Event structures arising from the unfolding of safe and finite Petri nets are equipped with a labelling of their events by transitions of the net. It is therefore natural to consider local randomizations of these event structures that are such that $p_x = p_{x'}$ whenever branching cells x and x' are isomorphic as labelled event structures. Finite safe Petri nets equipped with such local randomizations are called Markov nets; they generalize Markov chains to concurrent systems. We show in this section that branching cells provide the adequate concept of “local state” for Markov nets. In particular, we show that the classical Law of Large Numbers (LLN) for Markov chains properly generalizes to Markov nets, provided that the set of all equivalence classes of isomorphic branching cells is taken as state space for Markov nets. Such equivalence classes, called *dynamic clusters*, are introduced next.

Throughout this section, we assume that \mathcal{E} is a locally finite event structure arising from the unfolding of a finite safe Petri net \mathcal{N} . Although Assumption 1 is always satisfied by the unfolding of a safe and finite Petri net, this is not necessarily the case for local finiteness (Assumption 2). Local finiteness is an important restriction, although the class of safe nets with locally finite unfolding is strictly larger than the classes of free-choice or confusion-free nets.

Let M_0 denote the initial marking of \mathcal{N} . For v a finite configuration of \mathcal{E} , we denote by $m(v)$ the marking reached in \mathcal{N} after the action of configuration v . It is well known that, up to an isomorphism of labelled event structure, the future \mathcal{E}^v is the unfolding of net \mathcal{N} from the initial marking $m(v)$. Whence:

$$\forall v, v' \in \mathcal{V}_{\mathcal{E}}, \quad m(v) = m(v') \Rightarrow \mathcal{E}^v = \mathcal{E}^{v'}. \quad (17)$$

It makes thus sense to denote by \mathcal{E}^m the event structure that unfolds \mathcal{N} starting from the reachable marking m . Since the reachable markings are finitely many, the futures $\mathcal{E}^v = \mathcal{E}^{m(v)}$ are finitely many up to isomorphism of labelled event structures. Since each set of branching cells $\delta(v)$ is finite, it follows then from Def. 3 that branching cells of \mathcal{E} are finitely many, *up to an isomorphism of labelled event structures*.

Definition 6 (dynamic cluster). *An isomorphism class of branching cells is called a dynamic cluster of \mathcal{N} . We denote by Σ the (finite) set of dynamic clusters. Dynamic clusters are generically denoted by the boldface symbol \mathbf{s} . The equivalence class of branching cell x is denoted by $\langle x \rangle$.*

It is shown in the extended version [2] that, if the event structure is confusion-free, branching cells can be interpreted as the events of a new event structure, called *choice structure*. The set of dynamic clusters Σ is then a finite alphabet that labels the choice structure. Under certain conditions, the labelled event structure obtained is actually itself the unfolding of a safe Petri net, called the *choice net*. The interested reader is referred to [2] for further details.

Definition 7 (Markov net). *A Markov net is a pair $(\mathcal{N}, (p_{\mathbf{s}})_{\mathbf{s} \in \Sigma})$, where \mathcal{N} is a finite safe Petri net with locally finite unfolding, and $p_{\mathbf{s}}$ is a probability on the finite set $\Omega_{\mathbf{s}}$ for every $\mathbf{s} \in \Sigma$.*

Markov net $(\mathcal{N}, (p_{\mathbf{s}})_{\mathbf{s} \in \Sigma})$ induces a locally randomized event structure $(\mathcal{E}, (p_x)_{x \in X})$ (see Def. 4) by setting $p_x = p_{\langle x \rangle}$ for every branching cell $x \in X_{\mathcal{E}}$, whence a unique distributed probability \mathbb{P} on Ω (Th. 5 and Th. 6). Note that, if net \mathcal{N} is the product of two non interacting nets $\mathcal{N} = \mathcal{N}_1 \times \mathcal{N}_2$, then the two components $\mathcal{N}_i, i \in \{1, 2\}$ are independent in the probabilistic sense, i.e., $\mathbb{P} = \mathbb{P}_1 \otimes \mathbb{P}_2$.

Theorem 7 (Markov property). *Let $(\mathcal{N}, (p_{\mathbf{s}})_{\mathbf{s} \in \Sigma})$ be a Markov net, and let \mathbb{P} be the associated distributed probability on Ω . For v a finite recursively stopped configuration of \mathcal{E} , let $m(v)$ and Σ^v denote respectively the marking reached by v and the classes of branching cells of \mathcal{E}^v . Then for every $v \in \mathcal{W}$, the probabilistic future $(\mathcal{E}^v, \mathbb{P}^v)$ is associated with Markov net $(\mathcal{N}^v, (p_{\mathbf{s}})_{\mathbf{s} \in \Sigma^v})$, where \mathcal{N}^v is the same net as \mathcal{N} , except that \mathcal{N}^v has initial marking $m(v)$. Moreover we have:*

$$\forall v, v' \in \mathcal{W}, \quad m(v) = m(v') \Rightarrow \mathbb{P}^v = \mathbb{P}^{v'}. \quad (18)$$

Eqn. (18) expresses the memoryless nature of Markov nets: the probabilistic future of a $v \in \mathcal{W}$ only depends on the final marking $m(v)$. It is the probabilistic counterpart of Eqn. (17).

The Law of Large Numbers (LLN). Call *return* to the initial marking M_0 any finite recursively stopped configuration v such that:

1. $m(v) = M_0$,
2. $\text{Min}_{\preceq}(E) \cap \text{Min}_{\preceq}(E^v) = \emptyset$.

Informally, Point 2 above says that all the tokens in the net have moved when we apply configuration v . It prohibits recurrent behaviors that leave a part of the initial marking unchanged. For our study of LLN, we restrict ourselves to *recurrent* Markov nets, i.e., Markov nets such that, with probability 1, $\omega \in \Omega$ contains infinitely many returns to M_0 . If the considered net is indeed sequential, then our definition reduces to the classical notion of recurrence, for Markov chains [12].

For finite recurrent Markov chains, the LLN states as follows. Let Σ be the finite state space of a Markov chain $(X_k)_{k \geq 1}$, and let $f : \Sigma \rightarrow \mathbb{R}$ be a test function. The sums $S_n(f) = \sum_{k=1}^n f(X_k)$ are called *ergodic sums*, and the LLN studies the limit, for $n \rightarrow \infty$, of the *ergodic means*: $M_n(f) = \frac{1}{n} S_n(f)$. In extending the LLN to Markov net \mathcal{N} , we are faced with two difficulties:

1. What is the proper concept of state?
2. What replaces counter n , since time is not totally ordered?

Corresponding answers are:

1. The set Σ of dynamic clusters of \mathcal{N} is taken as the state space.
2. For v a recursively stopped configuration, the number of branching cells contained in $\Delta(v)$ is taken as the “duration” of v .

More precisely, call *distributed function* a finite family $f = (f_{\mathbf{s}})_{\mathbf{s} \in \Sigma}$ of real-valued functions $f_{\mathbf{s}} : \Omega_{\mathbf{s}} \rightarrow \mathbb{R}$. Distributed functions form a vector space of finite dimension over \mathbb{R} . The *concurrent ergodic sums* of f are defined as the function $S(f)$:

$$S(f) : \mathcal{W} \rightarrow \mathbb{R}, \quad \forall v \in \mathcal{W}, \quad S(f)(v) = \sum_{x \in \Delta(v)} f_{(x)}(v \cap x). \quad (19)$$

For example, if $N = (N_{\mathbf{s}})_{\mathbf{s} \in \Sigma}$ is the distributed function given by $N_{\mathbf{s}}(w) = 1$ for all $\mathbf{s} \in \Sigma$ and $w \in \Omega_{\mathbf{s}}$, then $S(N)(v)$ counts the number of branching cells contained in $\Delta(v)$. The *concurrent ergodic means* $M(f) : \mathcal{W} \rightarrow \mathbb{R}$ associated with a distributed function f are defined as the following ratios:

$$\forall v \in \mathcal{W}, \quad M(f)(v) = \frac{1}{S(N)(v)} S(f)(v). \quad (20)$$

The LLN is concerned by the limit

$$\lim_{v \subseteq \omega, v \rightarrow \omega} M(f)(v), \quad (21)$$

and this for each $\omega \in \Omega$, in a sense we shall make precise. The following notion of *stopping operator* will be central in this respect—stopping operators indeed generalize stopping times [12] for sequential stochastic processes:

Definition 8 (stopping operator). *A random variable $V : \Omega \rightarrow \mathcal{W}$, satisfying $V(\omega) \subseteq \omega$ for all $\omega \in \Omega$, is called a stopping operator if for all $\omega, \omega' \in \Omega$, we have: $\omega' \supseteq V(\omega) \Rightarrow V(\omega') = V(\omega)$. Say that a sequence $(V_n)_{n \geq 1}$ of stopping operators is regular if the following properties are satisfied—such sequences exist:*

1. $V_n \subseteq V_{n+1}$ for all n , and $\bigcup_n V_n(\omega) = \omega$ for all $\omega \in \Omega$;
2. there are two constants $k_1, k_2 > 0$ such that, with N the distributed function defined above, for all $\omega \in \Omega$ and all $n \geq 1$: $k_1 n \leq S(N)(V_n(\omega)) \leq k_2 n$.

Using this concept, Eqn. (21) is re-expressed as follows:

Definition 9 (convergence of ergodic means). For f a distributed function, we say that the ergodic means $M(f)$ converge to a function $\mu : \Omega \rightarrow \mathbb{R}$ if for every regular sequence $(V_n)_{n \geq 1}$ of stopping operators,

$$\lim_{n \rightarrow \infty} M(f)(V_n(\omega)) = \mu(\omega) \text{ with probability 1.} \quad (22)$$

Concurrency prevents property (22) from holding for general recurrent Markov nets, as the following particular case shows. Assume that net \mathcal{N} decomposes as $\mathcal{N} = \mathcal{N}_1 \times \mathcal{N}_2$ and the two components \mathcal{N}_1 and \mathcal{N}_2 do not interact at all. In this case, regular sequences $V = (V_n)_{n \geq 1}$ of stopping operators decompose into pairs (V^1, V^2) of independent regular sequences, one for each component. For f and v decomposed as $f = (f_1, f_2)$ and $v = (v_1, v_2)$ respectively, we have $S(f)(v) = S(f_1)(v_1) + S(f_2)(v_2)$ and $S(N)(v) = S(N_1)(v_1) + S(N_2)(v_2)$. Since V_n^1 and V_n^2 are free to converge at their own speed, we cannot expect that convergence of ergodic means will hold for this case. Clearly, concurrency is the very cause for this difficulty.

For the detailed statement of the condition needed to overcome this problem, the reader is referred to [1–Ch.8]. We only give an informal explanation, in terms of Petri nets and branching cells. If, in an execution $\omega \in \Omega$, we block a token represented by some condition b in the unfolding, we measure the “loss of synchronization” of the system by counting the number of branching cells that can be traversed *without moving the blocked token*. This length defines a random variable $\Omega \rightarrow \mathbb{R}$ for each condition b of the unfolding. We say that the considered Markov net has *integrable concurrency height* if all these random variables are integrable, i.e., possess finite expectation w.r.t. probability \mathbb{P} , for b ranging over the set of all conditions of the unfolding. Remark that, due to the memoryless property of the system, this set of random variables is actually finite.

Theorem 8 (Law of Large Numbers). Let $(\mathcal{N}, (p_{\mathbf{s}})_{\mathbf{s} \in \Sigma})$ be a Markov net. Assume that \mathcal{N} is recurrent and has integrable concurrency height. Then:

1. For any distributed function $f = (f_{\mathbf{s}})_{\mathbf{s} \in \Sigma}$, the ergodic means $M(f)$ converge in the sense of Def. 9 to a function $\mu(f) : \Omega \rightarrow \mathbb{R}$.
2. Except possibly on a set of zero probability, $\mu(f)$ is constant and given by:

$$\mu(f) = \sum_{\mathbf{s} \in \Sigma} p_{\mathbf{s}}(f_{\mathbf{s}}) \alpha(\mathbf{s}), \quad \text{with: } p_{\mathbf{s}}(f_{\mathbf{s}}) = \sum_{w \in \Omega_{\mathbf{s}}} f_{\mathbf{s}}(w) p_{\mathbf{s}}(w). \quad (23)$$

3. In formula (23), coefficients $\alpha(\mathbf{s})$ are equal to

$$\alpha(\mathbf{s}) = \mu(N^{\mathbf{s}}), \quad (24)$$

and satisfy $\alpha(\mathbf{s}) \in [0, 1]$ and $\sum_{\mathbf{s}} \alpha(\mathbf{s}) = 1$; $\alpha(\mathbf{s})$ is the asymptotic rate of occurrence of local state \mathbf{s} in a typical execution $\omega \in \Omega$.

Statement 3 is a direct consequence of statements 1 and 2: Fix $\mathbf{s} \in \Sigma$, and consider the distributed function $N^{\mathbf{s}}$ defined by $N^{\mathbf{s}}(w) = 1$ for all $w \in \Omega_{\mathbf{s}}$ and $N^{\mathbf{s}} = 0$ if $\mathbf{s} \neq \mathbf{s}'$. Applying statements 1 and 2 to $N^{\mathbf{s}}$ yields $\alpha(\mathbf{s}) = \mu(N^{\mathbf{s}})$. In particular, from $N = \sum_{\mathbf{s}} N^{\mathbf{s}}$ we obtain: $\sum_{\mathbf{s}} \alpha(\mathbf{s}) = 1$.

If the net is actually sequential (i.e., reduces to a recurrent finite Markov chain), then Σ is the state space of the chain and coefficients $\alpha(\mathbf{s})$ are equal to the coefficients of the invariant measure of the chain. This again reveals that dynamic clusters play the role of local states for concurrent systems.

5 Conclusion and Perspectives

We have proposed branching cells as a form of local concurrent state for prime event structures and safe Petri nets. Our study applies to so-called locally finite event structures that significantly extend the confusion-free case. We have applied this to probabilistic event structures: for \mathcal{E} an event structure with set of maximal configurations Ω , there is a one-to-one correspondence between local randomizations of the branching cells of \mathcal{E} on the one hand, and the class of distributed probabilities on Ω on the other hand. Distributed probabilities yield concurrent systems in which locally concurrent random choices are taken independently in the probabilistic sense.

We have applied the construction of distributed probabilities to unfoldings of safe and finite Petri nets. This leads to the model of Markov nets, a probabilistic model of concurrent system specified by finitely many parameters. Besides the relation between causal and probabilistic independence, Markov nets bring the Markov property as a probabilistic counterpart to the memoryless nature of Petri nets. The Law of Large Numbers extends to Markov nets, with dynamic clusters taken as states. Therefore branching cells and dynamic clusters provide the adequate notion of local state, for systems with concurrency.

Acknowledgments. We wish to thank Philippe Darondeau for fruitful discussions and hints.

References

1. Abbes, S.: *Probabilistic model for concurrent and distributed systems. Limit theorems and applications*. PhD Thesis (2004), IRISA-Université de Rennes 1.
2. Abbes, S. and Benvensite, A.: *Branching cells as local states for event structures and nets: probabilistic applications*. INRIA Research Report (2004) **RR-5347**. <http://www.inria.fr/rrrt/rr-5347.html>
3. Benveniste, A., Haar, S. and Fabre, E.: Markov nets: probabilistic models for distributed and concurrent systems. *IEEE Trans. on Aut. Cont.* **48**:11 (2003) 1936–1950.
4. Benveniste, A., Haar, S., Fabre, E. and Jard, C.: Distributed monitoring of concurrent and asynchronous systems. *Proc. of CONCUR'03, LNCS 2761* (2003), 1–26. Extended and improved version to appear in *Discrete Event Dynamic Systems: Theory and Application*, Kluwer, 2005.
5. Desel, J. and Esparza, R.: *Free choice Petri nets*. Cambridge University Press (1995).

6. Haas, P. J.: *Stochastic Petri nets*. Springer-Verlag (2002).
7. Hermanns, H., Herzog, U. and Katoen, J.-P.: Process algebra for performance evaluation. *T.C.S.* **274**:1 (2002) 43–88.
8. Larsen, K. G. and Skou, A.: Bisimulation through probabilistic testing. *Inf. and Comp.* **94**:1 (1991) 1–28.
9. Mateus, P., Mitchell, J. C. and Scedrov, A.: Composition of cryptographic protocols in a probabilistic polynomial-time process calculus. *Proc. of CONCURR'03, LNCS* **2761** (2003) 327–349.
10. Nielsen, M., Plotkin, G. and Winskel, G.: Petri nets, event structures and domains, part 1. *T.C.S.* **13** (1981) 85–108.
11. Segala, R. and Lynch, N.: Decision algorithms for probabilistic bisimulations. *Proc. of CONCUR'02, LNCS* **2421** (2002) 371–396.
12. Shiryayev, A. N.: *Probability*. Springer Verlag (1984).
13. Varacca, D., Völzer, H. and Winskel, G.: Probabilistic event structures and domains. *Proc. of CONCUR'04, LNCS* **3170** (2004) 481–496.
14. Völzer, H.: Randomized non-sequential processes. *Proc. of CONCUR'01, LNCS* **2154** (2001) 184–201.

A Appendix: Proof of Th. 4.

This section presents the proof of Th. 4 of Section 2. For the other proofs of results of Section 2, the reader is referred to the extended version [2]. For the proof of the Law of large numbers, we refer to [1].

Lemma 3. *If x, y are two distinct initial stopping prefixes, then $e \parallel f$ for all pairs $(e, f) \in x \times y$.*

Proof. Follows from the definitions, and from the fact that if x, y are two events in conflict, then there are two events x', y' in *minimal* conflict and with $x' \preceq x$ and $y' \preceq y$. \diamond

Proof of Th. 4. Remark first that $\delta(\emptyset)$ is finite. Indeed, choose for each $x \in \delta(\emptyset)$ an event e_x minimal in x . All $x \in \delta(\emptyset)$ are disjoint since they are minimal, hence all the e_x are distinct, and minimal in \mathcal{E} . Assumption 1 (applied with $v = \emptyset$) implies that they are finitely many, and thus $\delta(\emptyset)$ is finite. Assumption 2 implies that each $x \in \delta(\emptyset)$ is a finite prefix. It follows that thin prefixes B_ξ have $\bigcup_{x \in \delta(\emptyset)} x$ as finite upper bound.

Now let ξ be a subset of $\delta(\emptyset)$, and let $B_\xi = \bigcup_{x \in \xi} x$. For each configuration v of B_ξ , and for each $x \in \xi$, $v \cap x$ is clearly a configuration of x , whence a mapping: $\phi : \mathcal{V}_{B_\xi} \rightarrow \prod_{x \in \xi} \mathcal{V}_x$. For each tuple $(v_x)_{x \in \xi}$ with $v_x \in \mathcal{V}_x$, put $v = \bigcup_{x \in \xi} v_x$. Then v is clearly a prefix of B_ξ , and it follows from Lemma 3 that v is also conflict-free, thus v is a configuration of B_ξ . The mapping $(v_x)_{x \in \xi} \rightarrow v$ defined by this way is the inverse of ϕ , thus ϕ is a bijection. Clearly, ϕ maps the set of maximal configurations of B_ξ onto $\prod_{x \in \xi} \Omega_x$, which completes the proof.