Concurrency and Probability: Removing Confusion, Compositionally*

Roberto Bruni[1], Hernán Melgratti[2], and Ugo Montanari[1]

[1]Dipartimento di Informatica, Università di Pisa, Italy
[2]Departamento de Computación, Universidad de Buenos Aires - Conicet, Argentina

Abstract

Assigning a satisfactory truly concurrent semantics to Petri nets with confusion and distributed decisions is a long standing problem, especially if one wants to fully replace nondeterminism with probability distributions and no stochastic structure is desired/allowed. Here we propose a general solution based on a recursive, static decomposition of (finite, occurrence) nets in loci of decision, called *structural branching cells* (*s-cells*). Each s-cell exposes a set of alternatives, called *transactions*, that can be equipped with a general probabilistic distribution, i.e., as desired, nondeterminism can be entirely replaced by probability. The solution is formalised as a transformation from a given Petri net to another net whose transitions are the transactions of the s-cells and whose places are the places of the original net, with some auxiliary structure for bookkeeping. The resulting net is confusion-free, namely if a transition is enabled, then all its conflicting alternatives are also enabled. Thus sets of conflicting alternatives can be equipped with probability distributions, while nonintersecting alternatives are purely concurrent and do not introduce any nondeterminism: they are Church-Rosser and their probability distributions are independent. The validity of the construction is witnessed by a tight correspondence result with the recent approach by Abbes and Benveniste (AB) based on recursively stopped configurations in event structures. Some advantages of our approach over AB's are that: i) s-cells are defined statically and locally in a compositional way, whereas AB's *branching cells* are defined dynamically (by executing the event structure) and globally (by manipulating the entire event structure); ii) their recursively stopped configurations correspond to possible executions, but the existing concurrency is not made explicit. Instead, our resulting nets are equipped with an original concurrency structure exhibiting a so-called complete concurrency property.

*Work carried on in part while the third author was attending a Program on Logical Structures in Computation at Simons Institute, Berkeley, December 2016.

1 Introduction

Concurrency theory and practice provide a useful abstraction for the design and use of a variety of systems. Concurrent computations, as defined in many models, are equivalence classes of execution sequences, called *traces*, where the order of concurrent events is inessential. Sequences in the same class should be indistinguishable for the current purpose of interest.

Probabilistic programming is widespread in fields like security, approximate computing, machine learning, quantum computing. For example, it is shown in [13] that it is possible to equip ordinary programming languages with probabilistic choices in such a way that weakest precondition analysis can be used to prove important properties of probabilistic programs.

However, the interplay between concurrency and choices driven by general probabilistic distributions has proved hard to convey in one computational model, as recognised, e.g., in [10].

Petri nets [21] are a basic, well understood model of concurrency, which we assume the reader is familiar with (otherwise, see the short introduction in Section 2.2). To some extent, Petri nets stand to concurrent computation as automata to sequential computation. For Petri nets, deterministic processes (*processes* for short) [12], or, equivalently, configurations of event structures [19, 27] represent equivalence classes of firing sequences, i.e. traces. Due to their simplicity and extendability, we think that Petri nets offer the ideal playground to experiment in combining concurrency and probabilities, leaving space to later transferring the results over more articulate models and programming paradigms. The problem addressed in this paper is a foundational one: *can concurrency and general probabilistic distributions coexist in Petri nets? If so, under which circumstances?* For *coexistence* we mean that all the following issues must be addressed:

1. *Speed independence*: Truly concurrent semantics usually requires that the computation is time-independent and also independent from the relative speed of processes. In this sense, while attaching weights or rates of stochastic distributions to transitions is perfectly fine with interleaving semantics (where decisions are taken globally, on the basis of currently enabled transitions), they are not always appropriate when truly concurrent semantics is considered.

2. *Pure probabilistic computation*: Nondeterminism must be replaced entirely by probabilistic choices. This means that whenever two transitions are enabled, the choice to fire one instead of the other is either inessential (because they are concurrent) or is driven by a probabilistic distribution.

3. *Schedule independence*: In connection to the previous items, the probability distribution that drives the choice of a transition (both its probability and its alternatives) must not be affected by the execution of concurrent events, i.e., concurrent events must be driven by independent probability distributions.

4. *Complete concurrency*: It should be possible to partition the firing sequences in equivalence classes and establish a bijective correspondence between such classes and a suitable set of concurrent, deterministic processes. Moreover, given a process it must be possible to recover all its corresponding firing sequences.

5. *Sanity check #1*: All the firing sequences that are associated to the same process must carry the same probability, i.e., the probability of a concurrent computation must be independent from the order of execution.

6. *Sanity check #2*: The sum of the probabilities assigned to all possible processes must be 1.

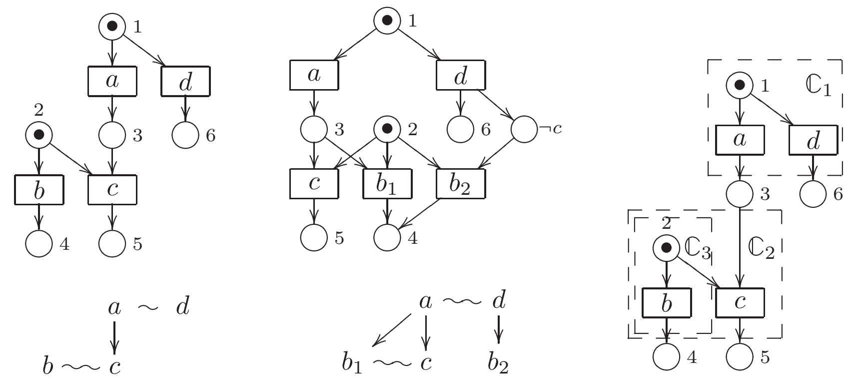
In this paper we provide a positive answer for the case of finite occurrence nets: given any such net we show how to define loci of decisions, called *structural branching cells* (*s-cells*), and construct another net where independent probability distributions can be assigned to concurrent events. At the implementation level this means that each s-cell can be assigned to a distributed random agent and that any concurrent computation is independent from the scheduling of agents.

The problem with confusion In the case of sequential systems, like automata, it is immediate to replace non-determinism with arbitrary probability distributions: they are assigned over all arcs (if any) leaving the same state.

The same happens when we move to *free-choice* Petri nets. There the presets of any two transitions are either disjoint or equal, thus the set of transitions with the same pre-set form a *cluster* that can be equipped with an arbitrary probability distribution. This is fine with concurrency, as a choice made in one cluster cannot change the set of alternatives available in another cluster, i.e., the probability distributions associated with any two clusters are independent. Note that being free-choice is a structural requirement: it is independent from the initial marking of the net.

By a similar reasoning, *confusion-free* Petri nets (see Section 2.2) are also amenable for the replacement of nondeterminism by probabilistic choices. *Confusion* arises when the set of alternatives to an enabled transition can be increased or decreased by the firing of an independent transition. Since in confusion-free nets the above does not happen, the set of alternatives can be equipped with probability distributions [25] and it can be shown that the ordinary notion of process coexists with such probabilities, in the sense that they meet all the items in our list of desiderata. Unfortunately, being confusion-free is not a structural property, because it depends on the initial marking. Consequently, determining the loci of decisions where to assign a probability distribution over the alternatives must be done by considering all the possible dynamic executions.

Confusion naturally arises in concurrent and distributed systems and is intrinsically present in problems involving mutual exclusion [23]. It has been recognised and studied from the beginning of net research [22], and to address it in a general and acceptable way can be considered as a main open problem for

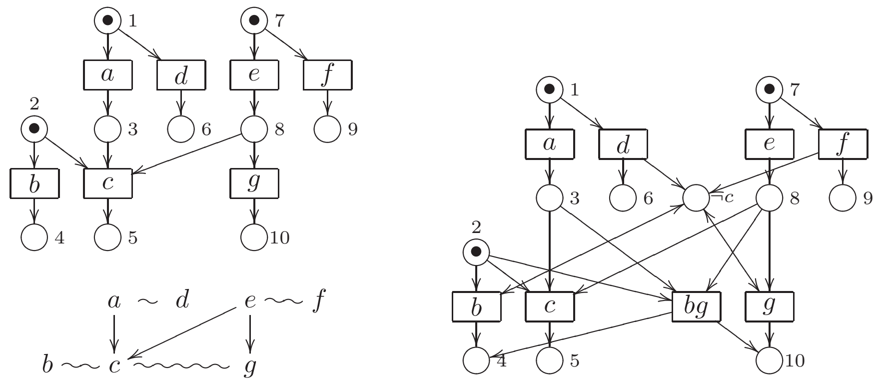


(a) Asymmetric confusion (b) Removing confusion (c) S-cells

Figure 1: A basic example: some nets (top) and their event structures (bottom)

concurrency theory. Citing [14], dealing with confusion, concurrency and probability all together is challenging. This is because confusion may break schedule independence and sanity checks. The simplest example of (asymmetric) confusion is the net in Fig. 1a. Apparently, transition b is concurrent w.r.t. a and d, but the firing of a enables c that is in conflict with b, while the firing of d definitively disables c, i.e., schedule independence is violated. Morally, there are two versions of b: one that is chosen in isolation and one that requires a choice between b and c (e.g. with probabilities p_b and $p_c = 1 - p_b$). However, from the concurrency point of view, there is a single process that comprises both a and b (as concurrent events), whose overall probability is hard to determine. If p_a is the probability of choosing a over d, then the trace $\sigma_1 = a; b$ has probability $p_a \cdot p_b$, while $\sigma_2 = b; a$ has probability $1 \cdot p_a = p_a$. This means that sanity check #1 fails for this process. Moreover, there are two other processes: one that comprises b and d as concurrent events (both its traces $b; d$ and $d; b$ have probability $p_d = 1 - p_a$) and one that comprises a and c (with a a cause of c), whose unique underlying trace $a; c$ has probability $p_a \cdot p_d = p_a \cdot (1 - p_b)$. From sanity check #2, we expect the sum of probabilities of all processes to be 1: this is the case if the process with a and b is assigned probability $p_a \cdot p_b$, i.e. if the trace σ_2 is not admissible.

As a general guideline, if the firing of a transition changes the set of alternatives available at some other site of the net, then it means that such transition is best executed before the choice at the other site happens, i.e., some causal dependency enforcing a suitable ordering of events must be added. According to this intuition, the idea is to delay the execution of b until all its potential alternatives have been enabled or disabled, so that the choice of firing b can be unambiguously equipped with different probability distributions in each case. In this sense, b should never be executed before a, because c can still be enabled.



(a) Confusion with OR-causes

(b) An attempt

Figure 2: Running example

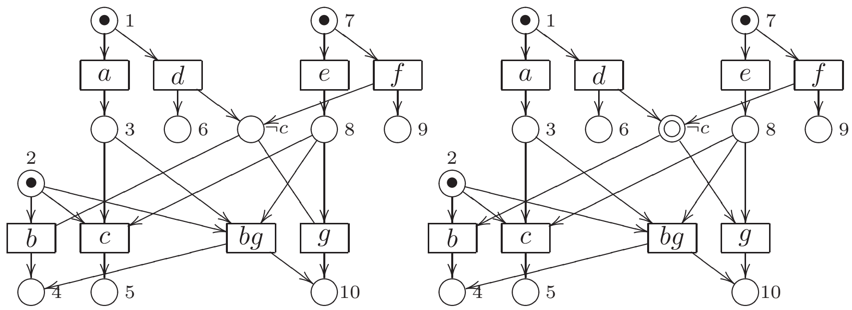
However, if d fires then c is discarded and b can be executed.

As a practical situation, imagine that a and d are the choices of your partner to either come to town (a) or go to the sea (d) and that you can go to the theatre alone (b), which is always an option, or go together with him/her (d), which is possible only when he/she is in town and accepts the invitation. Of course you better postpone the decision until you know if your partner is in town or not. This behaviour is faithfully represented, e.g., by the confusion-free net in Fig. 1b, where the two variants of b are made explicit (and named b_1 and b_2) and the new place $\neg c$ represents the information that c will never be enabled. Now, from the concurrency point of view, there is a single process that comprises both a and b_1 (with a a cause of b_1), whose overall probability is the product of the probability of choosing a instead of d by the probability of choosing b_1 over c. The other two processes comprise, respectively, d and b_2 (with d a cause of b_2) and a and c (with a a cause of c). As the net is confusion-free (although not free-choice) all criteria in the list of desiderata are met.

The general situation is much more complicated, because: i) there can be several ways to disable the same transition; ii) resolving a choice may require to execute more transitions at once. To see this, consider the net in Fig. 2a: i) c is discarded as soon as d *or* f fire; and ii) if both a and e fire we can choose to execute c alone or *both* b and g.

For example we can imagine that this time three persons are involved: Alice would like to play tennis with Carol, but they need Bob as a referee. Alice is already at the tennis court: she would like to play (c) but she can also practice alone (b); Bob and Carol can choose to go to the tennis court (a and e, respectively) or to stay home (d and f, respectively); if at the tennis court, Carol can also decide to practice alone (g).

Likewise the previous example, in this second, more general, scenario we



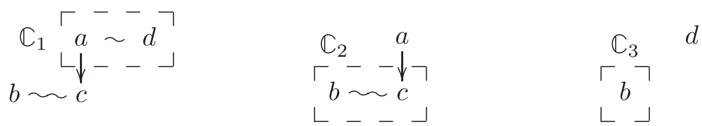
(a) An attempt with read arcs (b) A solution with persistent places

Figure 3: Running example continued

may expect to obtain a net like the one in Fig. 2b. Again, there is one place $\neg c$ to represent the disabling of c. This way a probability distribution can drive the choice between c and (the joint execution of) bg, whereas as soon as a token appears in $\neg c$ then b and g (if enabled) can fire concurrently.

A few things are worth to remark: i) the information stored in $\neg c$ can be used multiple times (from b and g), hence tokens in $\neg c$ should be read but not consumed (whence the double headed arcs from $\neg c$ to b and g, called *self-loops*); ii) multiple tokens can appear in the place $\neg c$ (by firing d and f, concurrently). These facts have severe repercussions on the concurrent semantics of the net. Suppose $d; f; b; g$ fire in this order: is the firing of b causally dependent on that of d or that of f (or on both)? Moreover, is the firing of g causally dependent on b (due to the self-loop b has on $\neg c$)? This last question can be solved if self-loops are replaced by *read arcs* [17] (see Fig. 3a), so that the firing of b does not alter the content of $\neg c$ and thus no causal dependency can arise between b and g. Nevertheless, if process semantics is considered, then we should explode all possible combinations of causal dependencies, thus introducing a new, undesired kind of nondeterminism. In reality, we should not expect any causal dependency between b and g, while both have OR dependencies on d and f.

To account for OR dependencies, we exploit the notion of *persistence*: tokens in a persistent place have infinite weight and are collective. Once a token reaches a persistent place, it cannot be removed and if two tokens reach the same persistent place they are indistinguishable. Such networks are not new: they are a variant of ordinary P/T nets and have been studied in [7]. In the example, we can declare $\neg c$ to be a persistent place and replace self-loops/read arcs on $\neg c$ with ordinary outgoing arcs (see Fig. 3b). Nicely we are able to introduce a process semantics for nets with persistent places that satisfies complete concurrency.



(a) Initial configuration (b) a is chosen (c) d is chosen

Figure 4: AB's dynamic branching cells for the example in Fig. 1a

Related Works. Confusion, as well as the interplay between concurrency and conflict, has been studied since longtime [22]. Moreover, recent emphasis on probabilistic models of computation has made the confusion problem more relevant. A number of probabilistic versions of Petri nets have been proposed [8, 15, 16, 11, 9]. Most of them replace nondeterminism with probability only in part, and furthermore introduce time dependent stochastic distributions, thus giving up the time and speed independence feature typical of proper concurrent models (our first item in the list). Confusion-free probabilistic models have been studied in [25], but this class, which subsumes free-choice nets, is usually considered quite restrictive.

Distributability of decisions has been studied, e.g., in [24, 14], but while the results in [24] apply to some restricted classes of nets, the approach in [14] requires nets to be decorated with agents and produces distributed models with both nondeterminism and probability, where concurrency is dependent from the scheduling of agents (i.e., it misses items 2–4 of our list).

A substantial advance in the study of concurrent, probabilistic models has been contributed by Abbes and Benveniste (AB) [1, 2, 3]. They consider prime event structures and suggest an elaborate construction which offers, for every maximal configuration, certain decompositions as alternating sequences of *branching cells* and *maximal stopping prefixes*. While the formal definition of the above concepts requires some effort (see Section 4.2), their result can be summarised as offering certain execution sequences where branching cells are loci of decision and maximal stopping prefixes are the extents of the computations associated to the choice. The event structure in Fig. 1a has three branching cells outlined in Fig. 4. First a decision between a and d must be taken: if a is executed, then a subsequent branching cell $\{b, c\}$ is enabled; otherwise the trivial branching cell $\{b\}$ is enabled. Branching cells are equipped with independent probability distributions. The probability assigned to a configuration via a given decomposition is the product of the probabilities assigned by its branching cells to its maximal stopping prefixes. Notably, the probability of a configuration does not depend on the chosen decomposition, and the sum of the probabilities attached to maximal configurations is 1. As expected, every decomposition of a configuration yields an execution sequence compatible with the configuration itself. Unfortunately, the converse does not hold in general:

certain sequences of events, legal w.r.t. the configuration, are not executable according to AB's decomposition. In this sense, AB's approach misses items 4 and 5 of our list of desiderata.

Contribution. In this paper we show how to systematically derive confusion-free nets (with persistency) from any (occurrence) Petri net and equip them with probabilistic distributions and concurrent semantics.

Technically, our approach is based on a structurally recursive decomposition of the original net in s-cells. As explained below, a simple kind of Asperti-Busi's dynamic nets is used as an intermediate model to structure the coding. While not strictly necessary, we feel that the intermediate step makes the construction more understandable: the translation from the given net to the dynamic net emphasises the hierarchical nature of the construction, while the second part is a general flattening step independent of our special case.

Our main results develop along two orthogonal axis.

1. Improvements over Abbes and Benveniste's construction.
   (a) Simplicity: s-cells definition in terms of a closure relation takes a couple of lines (see Definition 3), while AB's branching cell definition in terms of recursively stopped configurations in event structures requires some more advanced concepts and takes a couple of pages (we have done at our best to concisely recall it in Section 4.2).
   (b) Consistence: we define a correspondence between AB's branching cells and s-cells. Moreover, we relate AB's maximal configurations with our maximal processes, preserving their probability assignment.
   (c) Compositionality: s-cells are defined statically and locally, while AB's branching cells are defined dynamically (by executing the event structure) and globally (by manipulating the entire event structure).
   (d) Compilation vs interpretation: to some extent AB's construction gives an interpreter that rules out some executions of the event structure. Instead we compile the original net in another net (with persistency) whose execution is driven by ordinary firing rules.
   (e) Complete concurrency: AB's concurrent computations are the configurations of the event structure. Thus, in the presence of confusion, they include traces that are not executable by AB's interpreter, while our notion of process captures all and only executable traces.
2. Fully original perspectives.
   (a) Confusion removal: our target model is confusion-free by construction.
   (b) Locally executable model: our target model is extended as little as possible (i.e., with persistent places) to deal with OR dependencies arising from confusion, but for the rest it relies on ordinary firing

rules. Moreover, probabilistic choices are limited to transitions with the same pre-set, i.e., can be performed locally and concurrently.

(c) Processes: We define a novel notion of process for nets with persistency that conservatively extends the ordinary notion of process and captures the right amount of concurrency.

(d) Goal satisfaction: our construction meets all requirements in the list of desiderata.

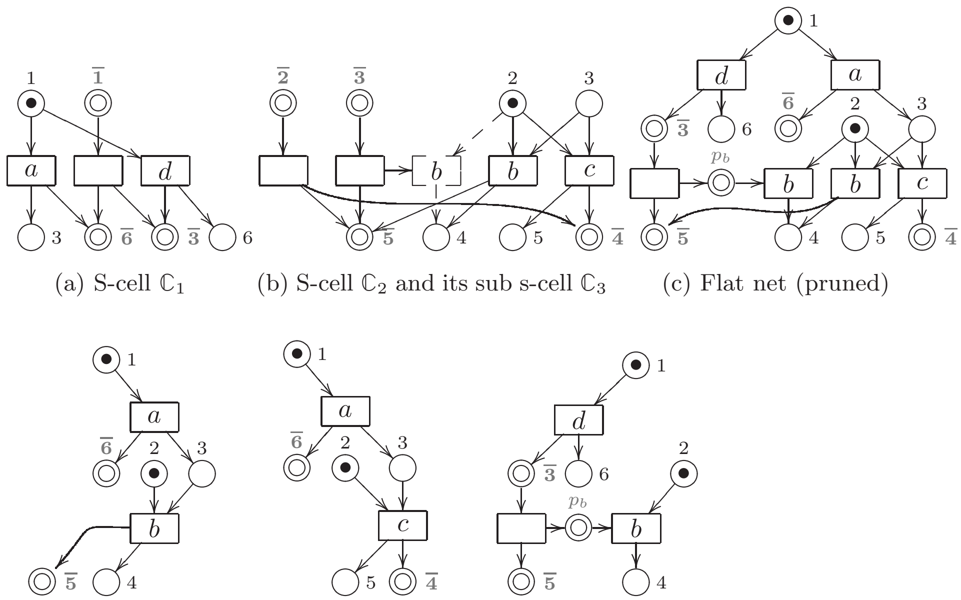
Structure of the paper After fixing notation in Section 2, our solution to the confusion problem consists of the following steps: (i) we define s-cells in a compositional way (Section 3.1); (ii) from s-cells decomposition we derive a confusion-free net with persistency (Section 3.2); (iii) we prove the correspondence with AB's approach (Section 4); (iv) we define a new notion of process that accounts for OR causal dependencies and meets the complete concurrency requirement (Section 5); and (iv) we show how to assign probabilistic distributions to s-cells (Section 6).

For (i), an s-cell $\mathbb{C}$ is uniquely determined by a set of transitions and defines a subnet $N_{\mathbb{C}}$. To account for the disappearance of alternatives from $\mathbb{C}$ caused by mutual exclusion, for each initial place p of $N_{\mathbb{C}}$ we recursively decompose the subnet $N_{\mathbb{C}} \ominus p$, where p and all its descendants are removed, in smaller s-cells.

About (ii), our target nets are built in two steps, by exploiting *dynamic* generation of transitions as a commodity and *persistency* of some places as a necessity.

First, to each s-cell there corresponds a *locally dynamic net*, whose transitions are the (atomic) maximal execution of the cell (in the vein of AB). Dynamic nets were introduced by Asperti and Busi [4, 5]: certain transitions of theirs are able to create (but not to remove) fresh places and transitions, thus acquiring great expressiveness. Here, *locally* dynamic nets can only create fresh transitions, not fresh places, and thus they are shown equivalent to static nets (see Section 2.5), since all transitions can be introduced in the initial net and just activated when needed. This is achieved in the second step.

Nets with persistency [7] (*p-nets*) are equipped with persistent places. Once marked, persistent places can be read any number of times. By construction, our p-nets have a peculiar structure. Each regular place p of our p-nets is paired with a corresponding persistent place $\overline{\mathbf{p}}$, meaning that, when marked, no token will ever mark p, i.e. we can say that p (and its descendants) *disappears*. In AB's approach, disappearance affects the parts of the event structure rejected in a choice point (see, e.g., the deletion of a and c while choosing d in the rightmost part of Fig. 4). It is easily defined, due to propagation of mutual exclusion through inheritance. However, to understand this step as a move of a transition system it is necessary to trace negative information through the net. We model this propagation via persistent places $\overline{\mathbf{p}}$. Intuitively, the transitions of a subnet $N_{\mathbb{C}} \ominus p$ are determined statically but released dynamically, when a token reaches $\overline{\mathbf{p}}$. Disappearance has a key role in our approach: a certain transition t, which is enabled but blocked due to a potential conflict with another transition u



(a) S-cell $\mathbb{C}_1$ (b) S-cell $\mathbb{C}_2$ and its sub s-cell $\mathbb{C}_3$ (c) Flat net (pruned)

(d) $a \preceq b$ (e) $a \preceq c$ (f) $d \preceq b$

Figure 5: S-cells, confusion-free p-net and processes for the net in Fig. 1a

(e.g., b w.r.t. c in Fig. 1a), may become unblocked as soon as the transition u is permanently disabled and thus disappears. Based on dynamic nets, in our approach the transition t is created when u disappears. Dynamic nets can then be flattened by adding an enabling persistent place $\mathbf{p}_t$ to any (dynamic) transition t.

A taste of the approach exemplified on the simplest case of asymmetric confusion is given next, while the second example (with OR dependencies) will serve as a step-by-step, running example.

A Taste of the Approach. We sketch the main ideas over the net in Fig. 1a. There are two main s-cells: $\mathbb{C}_1$ associated with $\{a, d\}$, and $\mathbb{C}_2$ with $\{b, c\}$. There is also a nested s-cell $\mathbb{C}_3$ that arises from the decomposition of the subnet $N_{\mathbb{C}_2} \ominus 3$. S-cells are shown in Fig. 1c. Their dynamic nets are in Figs. 5a–5b, where auxiliary transitions are in grey and unlabeled. Places $\overline{\mathbf{1}}$ and $\overline{\mathbf{2}}$ (and their transitions) are irrelevant, because the places 1 and 2 are already marked. However, our cells being static, we need to introduce auxiliary places in all cases. Note that in Fig. 5b there is an arc between two transitions. This is because the target transition is dynamically created when the other is executed (hence the dashed border). Also note that there are two transitions with the same label b: one is associated with the s-cell $\mathbb{C}_2$, the other with the unique s-cell $\mathbb{C}_3$ of $N_{\mathbb{C}_2} \ominus 3$ and is released when the place $\overline{\mathbf{3}}$ becomes marked.

After the s-cells are assembled and flattened we get the p-net in Fig. 5c (where irrelevant nodes are pruned). Initially, a and d are enabled. Firing a leads to the marking $\{2,3,\overline{\mathbf{6}}\}$ where $b : \{2,3\} \to \{4,\overline{\mathbf{5}}\}$ and $c : \{2,3\} \to \{\overline{\mathbf{4}},5\}$ are enabled (and in conflict). Firing d instead leads to the marking $\{2,\overline{\mathbf{3}},6\}$ where only the auxiliary transition can be fired, enabling $b : \{2,\mathbf{p}_b\} \to 4$. The net is confusion-free, as every conflict involves transitions with the same preset. For example, as the places 3 and $\overline{\mathbf{3}}$ (and thus $\overline{\mathbf{p}}_b$) are never marked in a same run, the transitions $b : \{2,3\} \to \{4,\overline{\mathbf{5}}\}$ and $b : \{2,\mathbf{p}_b\} \to 4$ will never compete for the token in 2.

Last, Figs. 5d–5f show the maximal processes of the net in Fig. 5c. It is evident that b can be executed neither before a nor before d (and the trace $b;a$ is not legal and not executable).

2 Preliminaries

2.1 Notation

We let $\mathbb{N}$ be the set of natural numbers, $\mathbb{N}_\infty = \mathbb{N} \cup \{\infty\}$ and $2 = \{0,1\}$. We write R^S for the set of functions from S to R: hence a subset of S is an element of 2^S, a multiset m over S is an element of $\mathbb{N}^S$, and a bag b over S is an element of $\mathbb{N}_\infty^S$, where elements of S can be assigned a finite or infinite multiplicity. By overloading the notation, union, difference and inclusion of sets, multisets and bags are all denoted by the same symbols: $\cup$, $\setminus$ and $\subseteq$, respectively. In the case of bags, the difference $b \setminus m$ is defined only when the second argument is a multiset, with the convention that $(b \setminus m)(s) = \infty$ if $b(s) = \infty$. Similarly, $(b \cup b')(s) = \infty$ if $b(s) = \infty$ or $b'(s) = \infty$. Sometimes we view a set as a multiset or a bag whose elements have unary multiplicity. Membership is denoted by $\in$: for a multiset m (or a bag b), we write $s \in m$ for $m(s) \neq 0$ ($b(s) \neq 0$).

Given a binary relation $R \subseteq S \times S$, we denote by R^+ its transitive closure and by R^* its reflexive and transitive closure. We say that R is *acyclic* if $\forall s \in S.\ (s,s) \notin R^+$.

2.2 Petri Nets, confusion and free-choiceness

A *net structure* N (also *Petri net*) [20, 21] is a tuple (P,T,F) where: P is the set of places, T is the set of transitions, and $F \subseteq (P \times T) \cup (T \times P)$ is the flow relation. For $x \in P \cup T$, we denote by ${}^\bullet x = \{y \mid (y,x) \in F\}$ and $x^\bullet = \{z \mid (x,z) \in F\}$ its *pre-set* and *post-set*, respectively. We assume that P and T are disjoint and non-empty and that ${}^\bullet t$ and $t^\bullet$ are non empty for every $t \in T$. We write $t : X \to Y$ for $t \in T$ with $X = {}^\bullet t$ and $Y = t^\bullet$.

A *marking* is a multiset $m \in \mathbb{N}^P$. We say that p is *marked* at m if $p \in m$. We write (N,m) for the net N *marked* by m. We write m_0 for the initial marking of the net, if any.

Graphically, a Petri net is a directed graph whose nodes are the places and transitions and whose set of arcs is F. Places are drawn as circles and transitions as rectangles. The current marking m is represented by inserting $m(p)$ tokens in each place $p \in m$ (see Fig. 1).

A transition t is *enabled* at the marking m, written $m \xrightarrow{t}$, if ${}^\bullet t \subseteq m$. The execution of a transition t enabled at m is called a *firing* and written $m \xrightarrow{t} m'$ with $m' = (m \setminus {}^\bullet t) \cup t^\bullet$. A firing sequence from m to m' is a finite sequence of firings $m = m_0 \xrightarrow{t_1} m_1 \xrightarrow{t_2} \cdots \xrightarrow{t_n} m_n = m'$, abbreviated to $m \xrightarrow{t_1 \cdots t_n} m'$ or just $m \to^* m'$. Moreover, $m \xrightarrow{t_1 \cdots t_n} m'$ is *maximal* if no transition is enabled at m'. We write $m \xrightarrow{t_1 \cdots t_n}$ if there is m' such that $m \xrightarrow{t_1 \cdots t_n} m'$. We say that m' is *reachable* from m if $m \to^* m'$. The set of markings reachable from m is written $[m\rangle$. A marked net (N, m) is *safe* if each $m' \in [m\rangle$ is a set. Safeness guarantees that $[m\rangle$ is finite (if P is finite) and that each place will contain at most one token at any time.

A net is called *free-choice* if for all transitions t, u we have either ${}^\bullet t = {}^\bullet u$ or ${}^\bullet t \cap {}^\bullet u = \emptyset$, i.e., if a transition t is enabled then all its conflicting alternatives are also enabled. So ${}^\bullet t$ is a locus of decision (not necessarily a singleton). Note that free-choiceness is purely structural. Confusion-freeness considers instead the dynamics of the net. A safe marked net (N, m_0) has *confusion* iff there exists a reachable marking m and transitions t, u, v such that:

symmetric case: (1) t, u, v are enabled at m, (2) ${}^\bullet t \cap {}^\bullet u \neq \emptyset \neq {}^\bullet u \cap {}^\bullet v$, (3) ${}^\bullet t \cap {}^\bullet v = \emptyset$; or

asymmetric case: (1) t and u are enabled at m, (2) v is not enabled at m but it becomes enabled after the firing of t, and (3) ${}^\bullet t \cap {}^\bullet u = \emptyset$ and ${}^\bullet v \cap {}^\bullet u \neq \emptyset$.

A net is *confusion-free* when it has no confusion.

Symmetric confusion means that there is the possibility to enable three transitions t, u, v such that the pre-sets of t and v are disjoint, but the firing of t changes the alternatives to v (because it disables u, which is in conflict with both t and v). Asymmetric confusion means that there is the possibility that the firing of a transition t concurrent with u enables an alternative v in conflict with u. An example of symmetric confusion is given by $m = \{2, 3, 8\}$, $t = b$, $u = c$ and $v = g$ in Fig. 2a, while an example of asymmetric confusion is given by $m = \{1, 2\}$, $t = a$, $u = b$ and $v = c$ in Fig. 1a.

2.3 Deterministic Nonsequential Processes

When tokens are seen as resources that are manipulated by transitions, we can address the issue of establishing when a firing is causally dependent on another one (because it removes some tokens produced by the other), or concurrent with it. This allows to consider equivalence classes of firing sequences of N up to a permutation of the order in which concurrent firings are executed. Such concurrent runs take the name of *deterministic nonsequential processes* [12].

A *deterministic nonsequential process* (or just *process*) represents the equivalence class of all firing sequences of a net that differ in the order in which concurrent firings are executed. It is given as a mapping $\pi : \mathcal{D} \to N$ from a *deterministic occurrence net* $\mathcal{D}$ to N (preserving pre- and post-sets), where a deterministic occurrence net is such that: (1) the flow relation is acyclic, (2) there are no backward conflicts ($\forall p \in P.\ |^{\bullet}p| \leq 1$), and (3) there are no forward conflicts ($\forall p \in P.\ |p^{\bullet}| \leq 1$). We let ${}^{\circ}\mathcal{D} = \{p \mid {}^{\bullet}p = \emptyset\}$ and $\mathcal{D}^{\circ} = \{p \mid p^{\bullet} = \emptyset\}$ be the sets of *initial* and *final places* of $\mathcal{D}$, respectively (with $\pi({}^{\circ}\mathcal{D})$ be the initial marking of N). When N is an acyclic safe net, the mapping $\pi : \mathcal{D} \to N$ is just an injective graph homomorphism: without loss of generality, we name the nodes in $\mathcal{D}$ as their images in N and let π be the identity.

The firing sequences of a processes $\mathcal{D}$ are its maximal firing sequences starting from the marking ${}^{\circ}\mathcal{D}$. Process $\mathcal{D}$ of N is *maximal* if its firing sequences are maximal in N.

For example, the net $\mathcal{D}$ with places $\{1, 2, 3, 4\}$ and transitions $\{a : 1 \to 3, b : 2 \to 4\}$ is a maximal process for the net in Fig. 1a. Its underlying (maximal) firing sequences are $m_0 \xrightarrow{a\ b}$ and $m_0 \xrightarrow{b\ a}$. We have ${}^{\circ}\mathcal{D} = \{1, 2\}$ and $\mathcal{D}^{\circ} = \{3, 4\}$.

Given an acyclic net we let $\preceq = F^*$ be the (reflexive) *causality* relation and say that two transitions t_1 and t_2 are in *immediate conflict*, written $t_1 \#_0 t_2$ if $t_1 \neq t_2 \ \wedge\ {}^{\bullet}t_1 \cap {}^{\bullet}t_2 \neq \emptyset$. The *conflict relation* $\#$ is defined by letting $x \# y$ if there are $t_1, t_2 \in T$ such that $(t_1, x), (t_2, y) \in F^+$ and $t_1 \#_0 t_2$. Then, a *nondeterministic occurrence net* (or just *occurrence net*) is a net $\mathcal{O} = (P, T, F)$ such that: (1) the flow relation is acyclic, (2) there are no backward conflicts ($\forall p \in P.\ |^{\bullet}p| \leq 1$), and (3) there are no self-conflicts ($\forall t \in T.\ \neg(t \# t)$). The *unfolding* $\mathcal{U}(N)$ of a safe Petri net N is an occurrence net that accounts for all (finite and infinite) runs of N: its transitions model all the possible instances of transitions in N and its places model all the tokens that can be created in any run. Our construction takes in input a finite occurrence net, which can be, e.g., the (truncated) unfolding of any safe net.

2.4 Nets With Persistency

Nets with persistency (*p-nets*) [7] partition the set of places in two parts, regular places P (ranged by $p, q, ...$) and persistent places $\mathbf{P}$ (ranged by $\mathbf{p}, \mathbf{q}, ...$). We use s to range over $\mathbb{S} = P \cup \mathbf{P}$ and write a p-net as a tuple $(\mathbb{S}, T, F)$. Intuitively, persistent places guarantee some sort of monotonicity about the knowledge of the system. Technically, this is realised by letting states be bags of places $b \in \mathbb{N}_{\infty}^{\mathbb{S}}$ instead of multisets, with the constraint that $b(p) \in \mathbb{N}$ for any regular place $p \in P$ and $b(\mathbf{p}) \in \{0, \infty\}$ for any persistent place $\mathbf{p} \in \mathbf{P}$. To guarantee that this property is preserved by firing sequences, we assume that the post-set $t^{\bullet}$ of a transition t is the bag such that: $(t^{\bullet})(p) = 1$ if $(t, p) \in F$ (as usual); $(t^{\bullet})(\mathbf{p}) = \infty$ if $(t, \mathbf{p}) \in F$; and $(t^{\bullet})(s) = 0$ otherwise. We say that a transition t is *persistent* if it is attached to persistent places only (i.e. if ${}^{\bullet}t \cup t^{\bullet} \subseteq \mathbf{P}$).

The notions of enabling, firing, firing sequence and reachability extend in the obvious way to p-nets (when markings are replaced by bags). For example, a transition t is *enabled* at the bag b, written $b \xrightarrow{t}$, if ${}^{\bullet}t \subseteq b$, and the firing of an enabled transitions is written $b \xrightarrow{t} b'$ with $b' = (b \setminus {}^{\bullet}t) \cup t^{\bullet}$. A firing sequence is *stuttering* if it has multiple occurrences of a persistent transition. Since firing a persistent transition t multiple times is inessential, we consider non-stuttering firing sequences. (Alternatively, we can add a marked regular place p_t to the preset of each persistent transition t, so t fires at most once.)

A marked p-net (N, b_0) is *1-∞-safe* if each reachable bag $b \in [b_0\rangle$ is such that $b(p) \in 2$ for all $p \in P$ and $b(\mathbf{p}) \in \{0, \infty\}$ for all $\mathbf{p} \in \mathbf{P}$. Note that in 1-$\infty$-safe nets the amount of information conveyed by any reachable bag is finite, as each place is associated with one bit of information (marked or unmarked). Graphically, persistent places are represented by circles with double border (and they are either empty or contain a single bullet). See Fig. 5c for an example.

The notion of confusion extends to p-nets, by checking conflicts w.r.t. regular places only.

2.5 Dynamic Nets

Dynamic nets [5] are Petri nets whose sets of places and transitions may increase dynamically. We focus on a subclass of dynamic nets that only allows changes in the set of transitions. In the following we let $\mathbb{S} = P \cup \mathbf{P}$ be a fixed set of regular places P and persistent places $\mathbf{P}$. The next definition introduces a version with persistency for such subclass.

Definition 1 (Dynamic p-nets)**.** *The set* $\textsc{dn}(\mathbb{S})$ *is the* least set *satisfying the recursive equation:*

$$\textsc{dn}(\mathbb{S}) = \{(T, b) \ | \ T \subseteq 2^{\mathbb{S}} \times \textsc{dn}(\mathbb{S}) \ \ \wedge \ \ T \textit{ finite} \ \wedge \ \ b \in \mathbb{N}_{\infty}^{\mathbb{S}}\}$$

The above recursive definition is a domain equation for the set of dynamic p-nets over the set of places $\mathbb{S}$: the set $\textsc{dn}(\mathbb{S})$ is the least fixed point of the equation. The simplest elements in $\textsc{dn}(\mathbb{S})$ are pairs $(\emptyset, b)$ with bag $b \in \mathbb{N}_{\infty}^{\mathbb{S}}$ (with $b(p) \in \mathbb{N}$ for any $p \in P$ and $b(\mathbf{p}) \in \{0, \infty\}$ for any $\mathbf{p} \in \mathbf{P}$). Nets (T, b) are defined recursively; indeed any element $t = (S, N) \in T$ stands for a transition with preset S and postset N, which is another element of $\textsc{dn}(\mathbb{S})$. An ordinary transition from b to b' has thus the form $(b, (\emptyset, b'))$. We write $S \to N$ for the transition $t = (S, N)$, ${}^\bullet t = S$ for its *preset*, and $t^\bullet = N \in \textsc{dn}(\mathbb{S})$ for its *postset*. For $N = (T, b)$ we say that T is the set of *top transitions* of N.

The firing rule rewrites a dynamic p-net (T, b) to another one. The firing of a transition $t = S \to (T', b') \in T$ consumes the preset S and releases both the transitions T' and the tokens in b'. Formally, if $t = S \to (T', b') \in T$ with $S \subseteq b$ then $(T, b) \xrightarrow{t} (T \cup T', (b \setminus S) \cup b')$.

It is evident that transitions are handled as persistent resources: they can be created but not erased. The notion of 1-∞-safe dynamic p-net is defined analogously to p-nets by considering the bags b of reachable states (T, b).

A sample dynamic net is in Fig. 5b, whose only dynamic transition has a dashed border.

We show that any dynamic p-net can be encoded as a (flat) p-net. Our encoding resembles the one in [5], but it is simpler because we do not need to handle place creation. Intuitively, we release any transition t immediately but we add a persistent place $\mathbf{p}_t$ to its preset, to enable t dynamically ($\mathbf{p}_t$ is initially

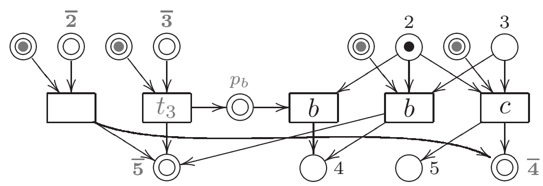


Figure 6: Flat net corresponding to the s-cell $\mathbb{C}_2$ in Fig. 5b

empty iff t is not a top transition). Given a set T of transitions, b_T is the bag such that $b_T(\mathbf{p}_t) = \infty$ if $t \in T$ and $b_T(s) = 0$ otherwise.

For $N = (T, b) \in \textsc{dn}(\mathbb{S})$, we let $\mathbb{T}(N) = T \cup \bigcup_{t \in T} \mathbb{T}(t^\bullet)$ be the set of all (possibly nested) transitions appearing in N. From Definition 1 it follows that $\mathbb{T}(N)$ is finite and well-defined.

Definition 2 (From dynamic to static)**.** *Given $N = (T, b) \in \textsc{dn}(\mathbb{S})$, the corresponding p-net $(\!|N|\!)$ is defined as $(\!|N|\!) = (\mathbb{S} \cup \mathbf{P}_{\mathbb{T}(N)}, \mathbb{T}(N), F, b \cup b_T)$, where*

- $\mathbf{P}_{\mathbb{T}(N)} = \{\mathbf{p}_t \mid t \in \mathbb{T}(N)\}$*; and*
- *F is such that for any $t = S \to (T', b') \in \mathbb{T}(N)$ then $t : {}^\bullet t \cup \{\mathbf{p}_t\} \to b' \cup b_{T'}$.*

The transitions of $(\!|N|\!)$ are those from N (set $\mathbb{T}(N)$). Any place of N is also a place of $(\!|N|\!)$ (set $\mathbb{S}$). In addition, there is one persistent place $\mathbf{p}_t$ for each $t \in \mathbb{T}(N)$ (set $\mathbf{P}_{\mathbb{T}(N)}$). The initial marking of $(\!|N|\!)$ is that of N (i.e., b) together with the persistent tokens that enable the top transitions of N (i.e., b_T). Adding b_T is convenient for the statement in Proposition 1, but we could safely remove $\mathbf{P}_T \subseteq \mathbf{P}_{\mathbb{T}(N)}$ (and b_T) from the flat p-net without any consequence.

Example 1. *Consider the dynamic net in Fig. 5b. It contains one dynamic transition, which is depicted with dashed lines. The corresponding static p-net is shown in Fig. 6, which has as many transitions as the original net. Note that the preset of any transition has been extended by adding a dedicated persistent place (depicted in grey) that indicates when the transition is available. Consequently, all the new places but p_b are initially marked to denote that the corresponding transitions are available. The place p_b instead is unmarked because the original dynamic transition is not initially available but it will become available when the transition t_3 is fired.*

The following result shows that the computations of a dynamic net can be mimicked by the corresponding p-net and vice versa. Consequently, the encoding preserves also 1-safety over regular places.

Proposition 1. *Let $N = (T, b) \in \textsc{dn}(\mathbb{S})$. Then,*

1. *$N \xrightarrow{t} N'$ implies $(\!|N|\!) \xrightarrow{t} (\!|N'|\!)$;*
2. *Moreover, $(\!|N|\!) \xrightarrow{t} N'$ implies there exists N'' such that $N \xrightarrow{t} N''$ and $N'' = (\!|N'|\!)$.*

Corollary 1. *$(\!|N|\!)$ is 1-∞-safe iff N is 1-safe.*

3 From Petri Nets to Dynamic P-Nets

In this section we show that any (finite, acyclic) net N can be associated with a confusion-free, dynamic p-net $[\![N]\!]$ by suitably encoding loci of decision. The mapping builds on the structural cell decomposition introduced below.

3.1 Structural Branching Cells

Roughly, a structural branching cell represents a statically determined locus of choice, where the firing of some transitions is considered against all the possible conflicting alternatives. To each transition t we want to assign a unique s-cell $[t]$. This is achieved by taking the equivalence class of t w.r.t. the equivalence relation $\leftrightarrow$ induced by the least preorder $\sqsubseteq$ that includes immediate conflict $\#_0$ and causality $\preceq$. As we want s-cells to represent loci of decision, we find it convenient to include in each s-cell $[t]$ also the places appearing in the pre-sets of the transitions in $[t]$, i.e., we let the relation Pre^{-1} be also included in $\sqsubseteq$, with $\mathsf{Pre} = F \cap (P \times T)$. This way, if we have an arc $(p, t) \in F$ then $p \sqsubseteq t$ because $p \preceq t$ and $t \sqsubseteq p$ because $(t, p) \in \mathsf{Pre}^{-1}$. Formally, we let $\sqsubseteq$ be the transitive closure of the relation $\#_0 \cup \preceq \cup \mathsf{Pre}^{-1}$. Since $\#_0$ is subsumed by the transitive closure of the relation $\preceq \cup \mathsf{Pre}^{-1}$, we can equivalently set $\sqsubseteq = (\preceq \cup \mathsf{Pre}^{-1})^*$. Then, we let $\leftrightarrow = \{(x, y) \mid x \sqsubseteq y \wedge y \sqsubseteq x\}$. Intuitively, the choices available in the equivalence class $[t]_{|\leftrightarrow}$ of a transition t must be resolved atomically.

Definition 3 (S-cells)**.** *Let $N = (P, T, F)$ be a finite, nondeterministic occurrence net. The set $\textsc{bc}(N)$ of* s-cells *is the set of equivalence classes of $\leftrightarrow$, i.e., $\textsc{bc}(N) = \{[t]_{|\leftrightarrow} \mid t \in T\}$.*

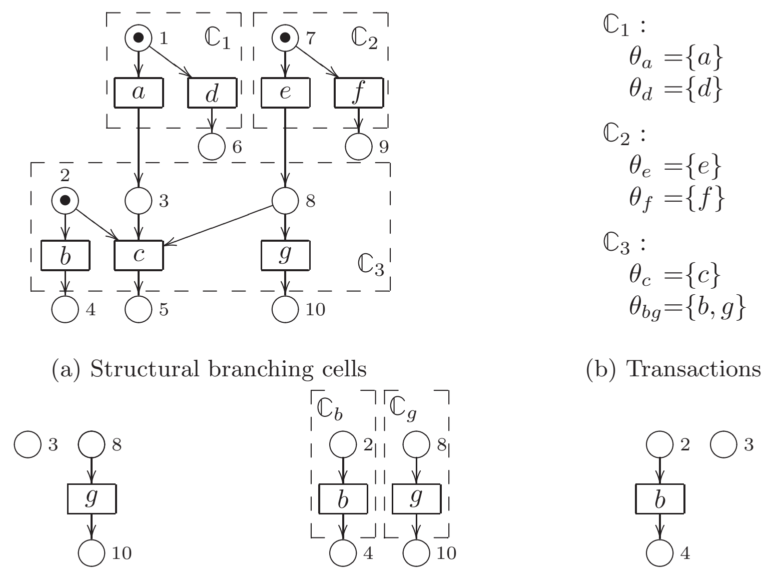
We let $\mathbb{C}$ range over s-cells. By definition it follows that for all $\mathbb{C}, \mathbb{C}' \in \textsc{bc}(N)$, if $\mathbb{C} \cap \mathbb{C}' \neq \emptyset$ then $\mathbb{C} = \mathbb{C}'$. For any s-cell $\mathbb{C}$, we denote by $N_{\mathbb{C}}$ the subnet of N whose elements are in $\mathbb{C} \cup \bigcup_{t \in \mathbb{C}} t^{\bullet}$. Abusing the notation, we denote by ${}^{\circ}\mathbb{C}$ the set of all the initial places in $N_{\mathbb{C}}$ and by $\mathbb{C}^{\circ}$ the set of all the final places in $N_{\mathbb{C}}$.

Definition 4 (Transactions)**.** *Let $\mathbb{C} \in \textsc{bc}(N)$. Then, a* transaction *$\theta$ of $\mathbb{C}$, written $\theta : \mathbb{C}$, is a maximal (deterministic) process of $N_{\mathbb{C}}$.*

Since the set of transitions in a transaction θ uniquely determines the corresponding process in $N_{\mathbb{C}}$, we write a transaction θ simply as the set of its transitions.

Example 2. *Consider the net N in Fig. 2a. It has the three s-cells shown in Fig. 7a, whose transactions are listed in Fig. 7b. For $\mathbb{C}_1$ and $\mathbb{C}_2$, each transition defines a transaction; $\mathbb{C}_3$ has one transaction associated with c and one with (the concurrent firing of) b and g.* □

The idea is that a s-cell $\mathbb{C}$ contains all the alternatives to choose from when the places in ${}^{\circ}\mathbb{C}$ are marked. However, at some point of the computation it can be the case that some of the places in ${}^{\circ}\mathbb{C}$ will never receive a token. In that case we may recursively decompose the s-cell $\mathbb{C}$ in smaller s-cells, by erasing the alternatives that are permanently disabled.



(a) Structural branching cells (b) Transactions

(c) $N_{\mathbb{C}_3} \ominus 2$ (d) $N_{\mathbb{C}_3} \ominus 3$ (e) $N_{\mathbb{C}_3} \ominus 8$

Figure 7: Structural branching cell decomposition (running example)

We use the operation $\ominus$ for the removal of a minimal place of a net and all the elements that causally depend on it. Formally, $N \ominus p$ is the least set that satisfies the rules (where $^\circ(\_)$ has higher precedence over set difference):

$$\frac{q \in {}^\circ N \setminus \{p\}}{q \in N \ominus p} \qquad \frac{t \in N \qquad {}^\bullet t \subseteq N \ominus p}{t \in N \ominus p} \qquad \frac{t \in N \ominus p \qquad q \in t^\bullet}{q \in N \ominus p}$$

Example 3. *Consider the s-cells in Fig. 7a. The net $N_{\mathbb{C}_1} \ominus 1$ is empty because every node in $N_{\mathbb{C}_1}$ causally depends on 1. Similarly, $N_{\mathbb{C}_2} \ominus 7$ is empty. The cases for $\mathbb{C}_3$ are in Figs. 7c–7e.* □

3.2 Encoding s-cells as confusion-free dynamic nets

The main idea is to explicitly represent the fact that a place will stay empty during the whole computation. We denote with $\overline{\mathbf{p}}$ the place that models such "negative" information about the regular place p and let $\overline{\mathbf{P}} = \{\overline{\mathbf{p}} \mid p \in P\}$.[1] The encoding uses negative information to recursively decompose branching cells under the assumption that some of their minimal places will stay empty.

Definition 5 (From s-cells to dynamic p-nets)**.** *Let $N = (P, T, F, m)$ be a marked occurrence net. Its dynamic p-net $[\![N]\!] \in \mathrm{DN}(P \cup \overline{\mathbf{P}})$ is defined as $[\![N]\!] = (T_{\mathrm{pos}} \cup T_{\mathrm{neg}}, m)$, where:*

[1] The notation $\overline{P}$ denotes just a set of places whose names are decorated with a bar; it should not be confused with usual set complement.

- $T_{\text{pos}} = \{\ {}^{\circ}\mathbb{C} \to (\emptyset, \theta^{\circ} \cup \overline{\mathbb{C}^{\circ} \setminus \theta^{\circ}})\ \mid\ \mathbb{C} \in \text{BC}(N)$ *and* $\theta : \mathbb{C}\ \}$, *and*

- $T_{\text{neg}} = \{\ \overline{\mathbf{p}} \to (T', \overline{\mathbb{C}^{\circ} \setminus (N_{\mathbb{C}} \ominus p)^{\circ}})\ \mid\ \mathbb{C} \in \text{BC}(N)$ *and* $p \in {}^{\circ}\mathbb{C}$ *and* $(T', b) = [\![N_{\mathbb{C}} \ominus p]\!]\ \}$.

For any s-cell $\mathbb{C}$ of N, and any transaction $\theta : \mathbb{C}$, the encoding contains a transition $t_{\theta,\mathbb{C}} = ({}^{\circ}\mathbb{C} \to (\emptyset, \theta^{\circ} \cup \overline{\mathbb{C}^{\circ} \setminus \theta^{\circ}})) \in T_{\text{pos}}$. Any such transition corresponds to the "atomic" execution of a maximal configuration of the s-cell, which contains the events associated with the firing of the transitions in θ. Despite ${}^{\circ}\theta$ may be strictly included in ${}^{\circ}\mathbb{C}$, we take ${}^{\circ}\mathbb{C}$ as the preset of $t_{\theta,\mathbb{C}}$, to ensure that the execution of θ only starts when the s-cell $\mathbb{C}$ is enabled. Each transition $t_{\theta,\mathbb{C}} \in T_{\text{pos}}$ is an ordinary Petri net transition because its postset consists of (i) the final places of θ and (ii) the persistent versions of the places in $\mathbb{C}^{\circ} \setminus \theta^{\circ}$. A token in $\overline{\mathbf{p}} \in \overline{\mathbb{C}^{\circ} \setminus \theta^{\circ}}$ represents the fact that the corresponding ordinary place $p \in \mathbb{C}^{\circ}$ will not be marked because it depends on discarded transitions (not in θ).

Negative information is propagated by the transitions in T_{neg}. For each cell $\mathbb{C}$ and place $p \in {}^{\circ}\mathbb{C}$, there exists one dynamic transition $t_{p,\mathbb{C}} = \overline{\mathbf{p}} \to (T', \overline{\mathbb{C}^{\circ} \setminus (N_{\mathbb{C}} \ominus p)^{\circ}})$ whose preset is just $\overline{\mathbf{p}}$ and whose postset propagates the negative information about p. Each transition in $t_{p,\mathbb{C}} \in T_{\text{neg}}$ is defined in terms of the subnet $N_{\mathbb{C}} \ominus p$. The postset for each transition in T_{neg} accounts for two effects of propagation: (i) the generation of negative tokens for all maximal places of $\mathbb{C}$ that causally depend on p, i.e., for the negative places associated with the ones in $\mathbb{C}^{\circ}$ that are not in $(N_{\mathbb{C}} \ominus p)^{\circ}$; and (ii) the activation of all transitions T' obtained by encoding $N_{\mathbb{C}} \ominus p$, which correspond to the behaviour of the branching cell $\mathbb{C}$ once the token in its minimal place p has been excluded. We remark that the bag b in $(T', b) = [\![N_{\mathbb{C}} \ominus p]\!]$ is always empty. Indeed: i) the net $N_{\mathbb{C}}$ is unmarked and, consequently, also $N_{\mathbb{C}} \ominus p$ is unmarked, and ii) the initial marking of $[\![N]\!]$ corresponds to the initial marking of N for any net.

Example 4. *Consider the net N and its s-cells in Fig. 7a. Then, $[\![N]\!] = (T, b)$, where the initial marking $b = \{1, 2, 7\}$ is the one of N and T has the transitions in Fig. 8.*

Consider the s-cell $\mathbb{C}_1$, with ${}^{\circ}\mathbb{C}_1 = \{1\}$, $\mathbb{C}_1^{\circ} = \{3, 6\}$ and transactions $\theta_a : \mathbb{C}_1$ and $\theta_d : \mathbb{C}_1$. In T_{pos} there is one transition for each transaction in $\mathbb{C}_1$, namely t_a and t_d. Both t_a and t_d have ${}^{\circ}\mathbb{C}_1$ as preset. Their postsets are nets with empty sets of transitions (by definition of T_{pos}): $t_a^{\bullet}$ produces the positive tokens $\theta_a^{\circ} = \{3\}$ and the negative ones in $\overline{\mathbb{C}_1^{\circ} \setminus \theta_a^{\circ}} = \overline{\{3, 6\} \setminus \{3\}} = \{\overline{\mathbf{6}}\}$, while $t_d^{\bullet}$ produces $\theta_d^{\circ} = \{6\}$ and $\overline{\mathbb{C}_1^{\circ} \setminus \theta_d^{\circ}} = \{\overline{\mathbf{3}}\}$. There is also one transition $t_1 \in T_{\text{neg}}$ for the propagation of the absence of tokens from ${}^{\circ}\mathbb{C}_1 = \{1\}$. Since $N_{\mathbb{C}_1} \ominus 1$ is the empty net, $[\![N_{\mathbb{C}_1} \ominus 1]\!] = (\emptyset, \emptyset)$. Hence, the postset of t_1 propagates negative tokens to all maximal places of $\mathbb{C}_1$, i.e., $\{\overline{\mathbf{3}}, \overline{\mathbf{6}}\}$.

For the s-cell $\mathbb{C}_2$ we get analogous transitions t_e, t_f and t_7.

The case of $\mathbb{C}_3$ is more interesting. It has ${}^{\circ}\mathbb{C}_3 = \{2, 3, 8\}$, $\mathbb{C}_3^{\circ} = \{4, 5, 10\}$ and transactions θ_{bg} and θ_c. Correspondingly, $[\![N]\!]$ has two transitions $t_{bg}, t_c \in T_{\text{pos}}$. Despite θ_{bg} mimics the firing of b and g that are disconnected from the place 3,

$$
\begin{array}{ll}
 & t_{bg} : 2,3,8 \to (\emptyset, \{4, 10, \overline{\mathbf{5}}\}) \quad \text{for } \theta_{bg} \\
 & t_c \ : 2,3,8 \to (\emptyset, \{5, \overline{\mathbf{4}}, \overline{\mathbf{10}}\}) \quad \text{for } \theta_c \\
t_a : 1 \to (\emptyset, \{3, \overline{\mathbf{6}}\}) \ \text{for } \theta_a & \\
t_d : 1 \to (\emptyset, \{6, \overline{\mathbf{3}}\}) \ \text{for } \theta_d & t_2 \ : \overline{\mathbf{2}} \to (\{t_g, t'_8\}, \{\overline{\mathbf{4}}, \overline{\mathbf{5}}\}) \\
t_1 : \overline{\mathbf{1}} \to (\emptyset, \{\overline{\mathbf{3}}, \overline{\mathbf{6}}\}) & t_3 \ : \overline{\mathbf{3}} \to (\{t_b, t'_2, t_g, t'_8\}, \{\overline{\mathbf{5}}\}) \\
 & t_8 \ : \overline{\mathbf{8}} \to (\{t_b, t'_2\}\{\overline{\mathbf{5}}, \overline{\mathbf{10}}\}) \\
t_e : 7 \to (\emptyset, \{8, \overline{\mathbf{9}}\}) \ \text{for } \theta_e & \quad \text{where:} \\
t_f : 7 \to (\emptyset, \{9, \overline{\mathbf{8}}\}) \ \text{for } \theta_f & \qquad t_b : 2 \to (\emptyset, \{4\}) \\
t_7 : \overline{\mathbf{7}} \to (\emptyset, \{\overline{\mathbf{8}}, \overline{\mathbf{9}}\}) & \qquad t'_2 : \overline{\mathbf{2}} \to (\emptyset, \{\overline{\mathbf{4}}\}) \\
 & \qquad t_g : 8 \to (\emptyset, \{10\}) \\
 & \qquad t'_8 : \overline{\mathbf{8}} \to (\emptyset, \{\overline{\mathbf{10}}\})
\end{array}
$$

Figure 8: Encoding of branching cells (running example)

the preset of t_{bg} *also includes* 3*, because the firing of the transaction is postponed until the choice in* $\mathbb{C}_1$ *is resolved.*

Transitions $t_2, t_3, t_8 \in T_{\mathrm{neg}}$ *propagate the negative information for the places in* ${}^\circ\mathbb{C}_3 = \{2,3,8\}$*. The transition* t_3 *has preset* ${}^\bullet t_3 = \{\overline{\mathbf{3}}\}$ *and its postset is derived from the net* $N_{\mathbb{C}_3} \ominus 3$*, which has two (sub) s-cells* $\mathbb{C}_b$ *and* $\mathbb{C}_g$ *(see Fig. 7d). The transitions* t_b *and* t'_2 *arise from* $\mathbb{C}_b$*, while* t_g *and* t'_8 *are obtained from* $\mathbb{C}_g$*. Hence,* $[\![N_{\mathbb{C}_3} \ominus 3]\!] = (\{t_b, t'_2, t_g, t'_8\}, \emptyset)$ *and* $t_{bg}^\bullet = (\{t_b, t'_2, t_g, t'_8\}, \{\overline{\mathbf{5}}\})$ *as* $\overline{\mathbb{C}_3^\circ \setminus (N_{\mathbb{C}_3} \ominus 3)^\circ} = \{\overline{\mathbf{5}}\}$*. Similarly, we derive* t_2 *from* $N_{\mathbb{C}_3} \ominus 2$ *and* t_8 *from* $N_{\mathbb{C}_3} \ominus 8$*.*

We now highlight some features of the encoded net. First, the set of top transitions is free-choice: positive and negative transitions have disjoint presets and the presets of any two positive transitions either coincide (if they arise from the same s-cell) or are disjoint. Recursively, this property holds at any level of nesting. Hence, the only source of potential confusion is due to the combination of transitions that are initially enabled and those that can be activated dynamically, e.g., t_b *and either* t_{bg} *or* t_c*. However,* t_b *is activated only when either* $\overline{\mathbf{3}}$ *or* $\overline{\mathbf{8}}$ *are marked, while* ${}^\bullet t_{bg} = {}^\bullet t_c = \{3, 8\}$*. Then, confusion is avoided by ensuring that* p *and* $\overline{\mathbf{p}}$ *are never marked in the same execution (Lemma 1).*

The net $[\![N]\!]$ *is shown in Fig. 9, where the places* $\{\overline{\mathbf{1}}, \overline{\mathbf{2}}, \overline{\mathbf{7}}\}$ *and the transitions* $\{t_1, t_7, t_2, t'_2\}$ *are omitted because superseded by the initial marking* $\{1, 2, 7\}$*.*

We remark that the same dynamic transition can be released by the firing of different transitions (e.g., t_b *by* t_3 *and* t_8*) and possibly several times in the same computation. Similarly, the same negative information can be generated multiple times. However this duplication has no effect, since we handle persistent tokens. For instance, the firing sequence* $t_d; t_f; t_3; t_8$ *releases two copies of* t_b *and marks* $\overline{\mathbf{5}}$ *twice. This is inessential for reachability, but has interesting consequences w.r.t. causal dependencies (see Section 5).* □

The remaining of this section is devoted to show that the encoding generates

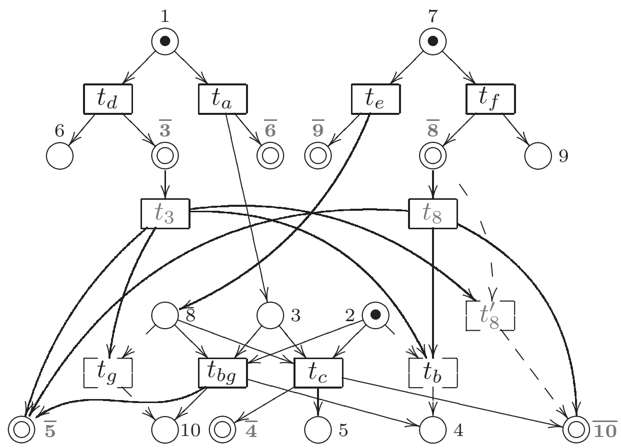


Figure 9: Dynamic net $[\![N]\!]$ (running example)

confusion-free nets. We start by stating a useful property of the encoded net, which is instrumental to the proof of the main result of the section.

Lemma 1 (Negative and positive tokens are mutually exclusive)**.** *If* $[\![N]\!] \to^* (T, b)$ *and* $\overline{\mathbf{p}} \in b$ *then* $(T, b) \to^* (T', b')$ *implies that* $p \notin b'$*.*

The above lemma ensures that no execution can generate tokens in both p and $\overline{\mathbf{p}}$. We now observe from Def. 5 that for any transition $t \in [\![N]\!] \in \mathrm{DN}(P \cup \overline{\mathbf{P}})$ it holds that either ${}^\bullet t \subseteq P$ or ${}^\bullet t \subseteq \overline{\mathbf{P}}$. The next result says that whenever there exist two transitions t and t' that have different but overlapping presets, at least one of them is disabled by the presence of a negative token in the marking b.

Lemma 2 (Nested rules do not collide)**.** *Let* $[\![N]\!] \in \mathrm{DN}(P \cup \overline{\mathbf{P}})$*. If* $[\![N]\!] \to^* (T, b)$ *then for all* $t, t' \in T$ *such that* ${}^\bullet t \neq {}^\bullet t'$ *and* ${}^\bullet t \cap {}^\bullet t' \cap P \neq \emptyset$ *it holds that there is* $p \in P \cap ({}^\bullet t \cup {}^\bullet t')$ *such that* $\overline{\mathbf{p}} \in b$*.*

The main result states that the encoding in Def. 5 generates confusion-free nets.

Theorem 1. *Let* $[\![N]\!] \in \mathrm{DN}(P \cup \overline{\mathbf{P}})$*. If* $[\![N]\!] \to^* (T, b) \xrightarrow{t}$ *and* $(T, b) \xrightarrow{t'}$ *then either* ${}^\bullet t = {}^\bullet t'$ *or* ${}^\bullet t \cap {}^\bullet t' = \emptyset$*.*

Corollary 2 (Confusion-free #1)**.** *Any net* $[\![N]\!] \in \mathrm{DN}(P \cup \overline{\mathbf{P}})$ *is confusion-free.*

Finally, we can combine the encoding $[\![\cdot]\!]$ with $(\!|\cdot|\!)$ (from Section 2.5) to obtain a (flat) 1-∞-safe, confusion-free, p-net $(\!|[\![N]\!]|\!)$, that we call the *uniformed net* of N. By Proposition 1 we get that the uniformed net $(\!|[\![N]\!]|\!)$ is also confusion-free by construction.

Corollary 3 (Confusion-free #2)**.** *Any p-net* $(\!|[\![N]\!]|\!)$ *is confusion-free.*

4 Static vs Dynamic cell decomposition

As mentioned in the Introduction, Abbes and Benveniste proposed a way of dynamically decomposing prime event structures in order to remove confusion from nets. In Sections 4.1 and 4.2 we recall the main notation and terminology needed to define prime event structures and branching cells, as introduced [1, 2, 3]. Then, we show that there is an operational correspondence between the dynamic decomposition of AB and the structural branching cells introduced in Section 3.1.

4.1 Prime Event Structures

A *prime event structure* (also *PES*) [18, 19, 27] is a triple $\mathcal{E} = (E, \preceq, \#)$ where: E is the set of *events*; $\preceq$ is a partial order on events called the *causality relation*; $\#$ is a symmetric, irreflexive relation on events called *conflict relation*; such that conflicts are inherited by causality, i.e., $\forall e_1, e_2, e_3 \in E.\ e_1 \# e_2 \preceq e_3 \Rightarrow e_1 \# e_3$.

The construction of the PES $\mathcal{E}_N$ associated with a net N can be formalised using category theory as a chain of universal constructions, called coreflections. As a consequence, to each PES $\mathcal{E}$, there is a standard, unique (up to isomorphism) nondeterministic occurrence net $N_{\mathcal{E}}$ that yields $\mathcal{E}$ and thus we can freely move from one setting to the other.

Consider the nets N in Figs. 1a and 2a. The corresponding PESs $\mathcal{E}_N$ are shown below each net. Events are in bijective correspondence with the transitions of the nets. Strict causality is depicted by arrows and immediate conflict by curly lines.

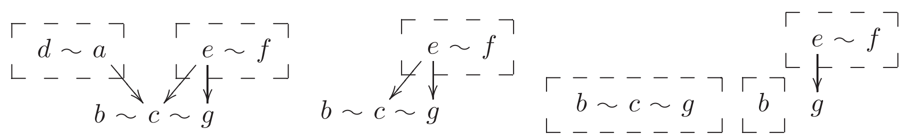
Given an event e, its *downward closure* $\lfloor e \rfloor = \{e' \in E \mid e' \preceq e\}$ is the set of causes of e. As usual, we assume that $\lfloor e \rfloor$ is finite for any e. Given $B \subseteq E$, we say that B is *downward closed* if $\forall e \in B.\ \lfloor e \rfloor \subseteq B$ and that B is *conflict-free* if $\forall e, e' \in B.\ \neg(e \# e')$. We let the *immediate conflict* relation $\#_0$ be defined on events by letting $e \#_0 e'$ iff $(\lfloor e \rfloor \times \lfloor e' \rfloor) \cap \# = \{(e, e')\}$, i.e., two events are in immediate conflict if they are in conflict but their causes are compatible.

4.2 Abbes and Benveniste's Branching Cells

In order to define AB's branching cells, some terminology must be introduced first. In the following we assume that a (finite) PES $\mathcal{E} = (E, \preceq, \#)$ is given.

A *prefix* $B \subseteq E$ is any downward-closed set of events (possibly with conflicts). Any prefix induces an event structure $\mathcal{E}_B = (B, \preceq_B, \#_B)$ where $\preceq_B$ and $\#_B$ are the restrictions of $\preceq$ and $\#$ to the events in B. A *stopping prefix* is a prefix B that is closed under immediate conflicts, i.e., $\forall e \in B, e' \in E.\ e \#_0 e' \Rightarrow e' \in B$. Intuitively, a stopping prefix is a prefix whose (immediate) choices are all available. It is *initial* if the only stopping prefix strictly included in B is $\emptyset$. We assume that any $e \in E$ is contained in a finite stopping prefix.

A *configuration* $v \subseteq \mathcal{E}$ is any set of events that is downward closed and conflict-free. Intuitively, a configuration represents (the state reached after executing) a concurrent but deterministic computation of $\mathcal{E}$. Configurations are



(a) $\mathcal{E}_N$ (b) $\mathcal{E}_N^{\{a\}}$ (c) $\mathcal{E}_N^{\{a,e\}}$ (d) $\mathcal{E}_N^{\{d\}}$

Figure 10: AB's branching cell decomposition (running example)

ordered by inclusion and we denote by $\mathcal{V}_\mathcal{E}$ the poset of finite configurations of $\mathcal{E}$ and by $\Omega_\mathcal{E}$ the poset of maximal configurations of $\mathcal{E}$.

The *future* of a configuration v, written E^v, is the set of events that can still be executed after v, i.e., $E^v = \{e \in E \setminus v \mid \forall e' \in v.\neg(e\#e')\}$. We denote by $\mathcal{E}^v$ the event structure induced by E^v. We assume that any finite configuration enables only finitely many events, i.e., that for any $v \in \mathcal{V}_\mathcal{E}$ the set of minimal elements in E^v w.r.t. $\preceq$ is finite.

A configuration v is *stopped* if there is a stopping prefix B with $v \in \Omega_B$. A configuration v is *recursively stopped* if there is a finite sequence of configurations $\emptyset = v_0 \subset v_1 \subset ... \subset v_n = v$ such that for any $i \in [0, n)$ the set $v_{i+1} \setminus v_i$ is a finite stopped configuration of the future $\mathcal{E}^{v_i}$ of v_i in $\mathcal{E}$.

A *branching cell* is any initial stopping prefix of the future $\mathcal{E}^v$ of a finite recursively stopped configuration v. Intuitively, a branching cell is a minimal subset of events closed under immediate conflict. We remark that branching cells are determined by considering the whole (future of the) event structure $\mathcal{E}$ and they are recursively computed as $\mathcal{E}$ is executed.

Every maximal configuration has a branching cell decomposition.

Example 5. *Consider the PES $\mathcal{E}_N$ in Fig. 2a and its maximal configuration $v = \{a, e, b, g\}$. We show that v is recursively stopped by exhibiting a branching cell decomposition of v. The initial stopping prefixes of $\mathcal{E}_N = \mathcal{E}_N^\emptyset$ are shown in Fig. 10a. There are two possibilities for choosing $v_1 \subseteq v$ and v_1 recursively stopped: either $v_1 = \{a\}$ or $v_1 = \{e\}$. When $v_1 = \{a\}$, the choices for v_2 are determined by the stopping prefixes of $\mathcal{E}_N^{\{a\}}$ (see Fig. 10b) and the only possibility is $v_2 = \{a, e\}$. From $\mathcal{E}_N^{\{a,e\}}$ in Fig. 10c, we take $v_3 = v$. Note that the configuration $\{a, e, b\}$ is not recursively stopped because $\{b\}$ is not maximal in the stopping prefix of $\mathcal{E}_N^{\{a,e\}}$ (see Fig. 10c). Finally, note that the branching cells $\mathcal{E}_N^{\{a\}}$ (Fig. 10b) and $\mathcal{E}_N^{\{d\}}$ (Fig. 10d) correspond to different alternatives in $\mathcal{E}_N^\emptyset$ and thus have different stopping prefixes.* $\square$

4.3 Operational correspondence between s-cells and AB's decomposition

We now establish the correspondence between s-cell decomposition of a given net N and the recursively-stopped configurations of (the event structure of) N

(i.e., of $\mathcal{E}_N$). We remark that the recursively-stopped configurations of a net N characterise the allowed executions of the net. Hence, we formally link the recursively-stopped configurations of $\mathcal{E}_N$ with the computations of the uniformed net $(\!|[\![N]\!]|\!)$.

Technically, we first show that the recursively-stopped configurations of $\mathcal{E}_N$ are in one-to-one correspondence with the computations of the dynamic net $[\![N]\!]$, which is more convenient to handle in the proofs than $(\!|[\![N]\!]|\!)$. Then, the desired correspondence is obtained by using the Proposition 1, which relates the computations of any dynamic net and its associated p-net.

We rely on the auxiliary map $\|-\|$ that allows us to link transitions in $[\![N]\!]$ with events in $\mathcal{E}_N$. More specifically, the mapping $\|-\|$ associates each transition t in $[\![N]\!]$ with the set $\|t\|$ of transitions in the original net N (i.e., events in $\mathcal{E}_N$) that are encoded by t. Formally,

$$\|t\| = \begin{cases} ev(\theta) & \text{if } t = t_{\theta,\mathbb{C}} \in T_{\text{pos}} \quad \text{where } ev(\theta) \text{ is the set of transitions in } \theta \\ \emptyset & \text{if } t \in T_{\text{neg}} \end{cases}$$

A transition t in $[\![N]\!]$ that is defined in terms of a transaction θ of some s-cell $\mathbb{C}$ is mapped to the set $ev(\theta)$ of transitions in θ. On the contrary, any transition in $[\![N]\!]$ that propagates negative information, i.e., $t \in T_{\text{neg}}$, is mapped to the empty set because it does not represent any firing of the original net N.

Example 6. *Consider the net N in Fig. 7a and its corresponding dynamic net defined by the rules in Fig. 8. The auxiliary mapping $\|\_\|$ is as follows*

$$\begin{array}{llll} \|t_a\| = \{a\} & \|t_d\| = \{d\} & \|t_e\| = \{e\} & \|t_f\| = \{f\} \\ \|t_{bg}\| = \{b, g\} & \|t_c\| = \{c\} & \|t_b\| = \{b\} & \|t_g\| = \{g\} \\ \multicolumn{4}{l}{\|t\| = \emptyset \textit{ if } t \in \{t_1, t_7, t_2, t_3, t_8, t_2', t_8'\}} \end{array}$$

Any transition $t_{\theta,\mathbb{C}}$ generated because of a transaction $\theta : \mathbb{C}$ is mapped to the set of transitions in θ. For instance, t_a is generated because of the transaction θ_a, which consists only on the transition a. Differently, transitions that propagate negative information, i.e., $t \in \{t_1, t_7, t_2, t_3, t_8, t_2', t_8'\}$, are mapped to the empty set because they are instrumental steps that do not encode the firing of any transition in the original net.

In what follows we write $M \Longrightarrow M'$ for a possibly empty firing sequence $M \xrightarrow{t_1 \cdots t_n} M'$ such that $\|t_i\| = \emptyset$ for all $i \in [1, n]$. If $\|t\| \neq \emptyset$, we write $M \stackrel{t}{\Longrightarrow} M'$ if $M \Longrightarrow M_0 \stackrel{t}{\Longrightarrow} M_1 \Longrightarrow M'$ for some M_0, M_1. Moreover, we write $M \stackrel{t_1 \cdots t_n}{\Longrightarrow}$ if there exist $M_1, ..., M_n$ such that $M \stackrel{t_1}{\Longrightarrow} M_1 \stackrel{t_2}{\Longrightarrow} \cdots \stackrel{t_n}{\Longrightarrow} M_n$.

The following result states that the computations of any dynamic net produced by the encoding $[\![\_]\!]$ are in one-to-one correspondence with the recursively-stopped configurations of Abbes and Benveniste.

Lemma 3 (Correspondence between $[\![\_]\!]$ and AB)**.** *Let N be an occurrence net.*

1. *If* $[\![N]\!] \xrightarrow{t_1 \cdots t_n}$*, then* $v = \bigcup_{1\le i\le n} \|t_i\|$ *is recursively-stopped in* $\mathcal{E}_N$ *and* $(\|t_i\|)_{1\le i\le n}$ *is a valid decomposition of* v*.*
2. *If* v *is recursively-stopped in* $\mathcal{E}_N$*, then for any valid decomposition* $(v_i)_{1\le i\le n}$ *there exists* $[\![N]\!] \xrightarrow{t_1 \cdots t_n}$ *such that* $\|t_i\| = v_i$*.*

Example 7. *Consider the branching cell decomposition for* $v = \{a, e, b, g\} \in \mathcal{E}_v$ *discussed in Ex. 5. Then, the net* $[\![N]\!]$ *in Ex. 4 can mimic that decomposition with the following computation* $(T, \{1, 2, 7\}) \xrightarrow{t_a} (T, \{2, 3, 7, \overline{\mathbf{6}}\}) \xrightarrow{t_e} (T, \{2, 3, 8, \overline{\mathbf{6}}, \overline{\mathbf{9}}\}) \xrightarrow{t_{bg}} (T, \{4, 10, \overline{\mathbf{5}}, \overline{\mathbf{6}}, \overline{\mathbf{9}}\})$ *where* $\|t_a\| = \{a\} = v_1$*,* $\|t_e\| = \{e\} = v_2$*, and* $\|t_{bg}\| = \{b, g\} = v_3$*.* □

The correspondence between the uniformed net and the recursively-stopped configurations associated with a net N is obtained by combining the previous result with Proposition 1.

Theorem 2 (Correspondence between $(\![[\![\_]\!]]\!)$ and AB)**.** *Let* N *be an occurrence net.*

1. *If* $(\![[\![N]\!]]\!) \xrightarrow{t_1 \cdots t_n}$*, then* $v = \bigcup_{1\le i\le n} \|t_i\|$ *is recursively-stopped in* $\mathcal{E}_N$ *and* $(\|t_i\|)_{1\le i\le n}$ *is a valid decomposition of* v*.*
2. *If* v *is recursively-stopped in* $\mathcal{E}_N$*, then for any valid decomposition* $(v_i)_{1\le i\le n}$ *there exists* $(\![[\![N]\!]]\!) \xrightarrow{t_1 \cdots t_n}$ *such that* $\|t_i\| = v_i$*.*

5 Complete Concurrency

In this section we study the amount of concurrency still present in the uniformed net $(\![[\![N]\!]]\!)$. Concurrency in the original net N is well understood: a deterministic process or an event structure configuration define precisely a concurrent computation corresponding to an equivalence class of firing sequences. However, in the presence of confusion, certain firing sequences that are legal in the configuration should not be executable. Here, we extend the notion of a process to the case of 1-∞-safe p-nets and we show that all the *legal* firing sequences of a process of the uniformed net $(\![[\![N]\!]]\!)$ are executable.

The notion of deterministic occurrence net is extended to p-nets by slightly changing the notions of conflict and of causal dependency: (i) two transitions with intersecting presets are not in conflict if all shared places are persistent, (ii) a persistent place can have more than one immediate cause in its preset, which will introduce OR-dependencies.

Definition 6 (Persistent process)**.** *An* occurrence p-net $O = (P \cup \mathbf{P}, T, F)$ *is an acyclic p-net such that* $|p^\bullet| \le 1$ *and* $|{}^\bullet p| \le 1$ *for any* $p \in P$ *(but not necessarily for those in* $\mathbf{P}$*).*

A persistent process *for* N *is then an occurrence p-net* O *together with a net morphism* $\pi : O \to N$ *that preserves presets and postsets of transitions and the distinction between regular and persistent places. Without loss of generality,* *when N is acyclic, we assume that O is just a subnet of N (with the same initial marking) and that π is the identity.*

In an ordinary occurrence net, the causes of an item x are all its predecessors. In p-nets, we want to model alternative sets of causes and thus the causes of an item x are represented as a formula $\Phi(x)$ of the propositional calculus without negation, where the basic propositions are the transitions of the occurrence net. If we represent such a formula as a sum of products, it corresponds to a set of *collections*, i.e. a set of sets of transitions. Different collections correspond to alternative causal dependencies, while transitions within a collection are all the causes of that alternative and *true* represents the empty collection. Such a formula $\Phi(x)$ represents a monotone boolean function, which expresses, as a function of the occurrences of past transitions, if x has enough causes. It is known that such formulas, based on positive literals only, have a unique DNF (sum of products) form, given by the set of prime implicants. In fact, every prime implicant is also essential [26]. We define $\Phi(x)$ by well-founded recursion:

$$\Phi(x) = \begin{cases} \mathit{true} & \text{if } x \in P \cup \mathbf{P} \wedge {}^{\bullet}x = \emptyset \\ \bigvee_{t \in {}^{\bullet}x} (t \wedge \Phi(t)) & \text{if } x \in P \cup \mathbf{P} \wedge {}^{\bullet}x \neq \emptyset \\ \bigwedge_{s \in {}^{\bullet}x} \Phi(s) & \text{if } x \in T \end{cases}$$

Ordinary deterministic processes satisfy *complete concurrency*: each process determines a partial ordering of its transitions, such that the executable sequences of transitions are exactly the linearizations of the partial order. More formally, after executing any firing sequence σ of the process, a transition t is enabled if and only if all its predecessors in the partial order (namely its causes) already appear in σ. In the present setting a similar property holds.

Definition 7 (Legal firing sequence)**.** *A sequence of transitions $t_1; \cdots ; t_n$ of a persistent process is* legal *if for all $k \in [1, n]$ we have that $\bigwedge_{i=1}^{k-1} t_i$ implies $\Phi(t_k)$.*

It is immediate to notice that if the set of persistent places is empty ($\mathbf{P} = \emptyset$), then the notion of persistent process is the ordinary one, $\Phi(x)$ is just the conjunction of the causes of x and a sequence is legal iff it is a linearization of the process. In this sense, persistent processes are a conservative extension of ordinary ones.

Theorem 3 (Complete Concurrency)**.** *Let $\sigma = t_1; \cdots ; t_n$ with $n \geq 0$ be a, possibly empty, firing sequence of a persistent process, and t a transition not in σ. The following conditions are all equivalent: (i) t is enabled after σ; (ii) there is a collection of causes of t which appears in σ; (iii) $\bigwedge_{i=1}^{n} t_i$ implies $\Phi(t)$.*

Corollary 4. *Given a persistent process, a sequence is legal iff it is a firing sequence.*

Example 8. *Fig. 11 shows a process for the net $(\!|[\![N]\!]|\!)$ of our running example (see N in Fig. 2a and $[\![N]\!]$ in Fig. 9). The process accounts for the firing of the transitions d, f, b in N. Despite they look as concurrent events in N, the persistent place $\mathbf{p_{t_b}}$ introduces some causal dependencies. In fact, we have: $\Phi(t_d) = \Phi(t_f) = \mathit{true}$, $\Phi(t_3) = t_d$, $\Phi(t_8) = t_f$ and $\Phi(t_b) = (t_3 \wedge t_d) \vee (t_8 \wedge t_f)$, thus t_b can be fired only after either t_d or t_f (or both).* □

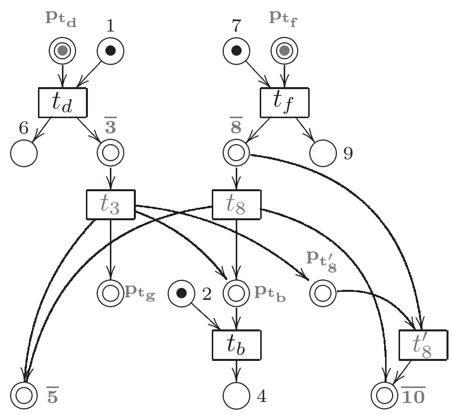


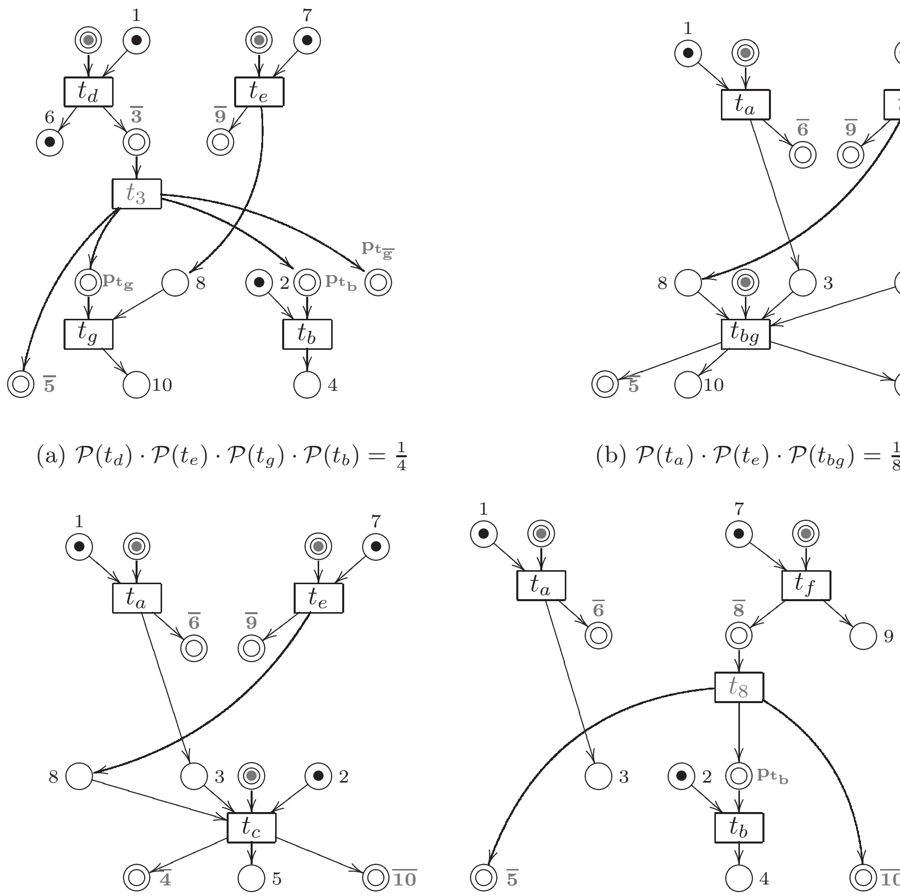
Figure 11: A process for $(\![\![N]\!]\!)$ (running example)

6 Probabilistic Nets

Building on the results from previous sections, we can now outline our methodology to assign probabilities to the concurrent runs of a Petri net, also in the presence of confusion. Given a net N, we apply s-cell decomposition from Section 3.1, and then we assign probability distributions to the transactions available in each cell $\mathbb{C}$ (and recursively to the s-cell decomposition of $N_{\mathbb{C}}$): let $\mathcal{P}_{\mathbb{C}} : \{\theta \mid \theta : \mathbb{C}\} \to [0,1]$ denote the probability distribution function of the s-cell $\mathbb{C}$ (such that $\sum_{\theta:\mathbb{C}} \mathcal{P}_{\mathbb{C}}(\theta) = 1$). Such probability distributions are defined purely locally and transferred automatically to the transitions in T_{pos} of the dynamic p-net $[\![N]\!]$ defined in Section 3, in such a way that $\mathcal{P}(t_{\theta,\mathbb{C}}) = \mathcal{P}_{\mathbb{C}}(\theta)$. Negative transitions in T_{neg} are each assigned probability 1 as no choice is associated with them. Since the transitions of the uniformed net $(\![\![N]\!]\!)$ are the same as $[\![N]\!]$, the probability distribution can be carried over $(\![\![N]\!]\!)$ (thanks to Proposition 1).

AB's probability distribution Building on the bijective correspondence in Theorem 3, the distribution $\mathcal{P}_{\mathbb{C}}$ can be chosen in such a way that it is consistent with the one attached to the transitions of Abbes and Benveniste's branching cells (if any).

Purely local distribution Another way to define $\mathcal{P}_{\mathbb{C}}$ is by assigning probability distributions to the arcs leaving the same place, as if each place were able to decide autonomously which transition to fire. Then, given a transaction $\theta : \mathbb{C}$, we can set $\mathcal{Q}_{\mathbb{C}}(\theta)$ be the product of the probability associated with the arcs of N entering the transitions in θ. Of course, in general it can happen that $\sum_{\theta:\mathbb{C}} \mathcal{Q}_{\mathbb{C}}(\theta) < 1$, as not all combinations of token flows correspond to maximal processes. However, it is always possible to normalise the quantities of feasible assignments by setting $\mathcal{P}_{\mathbb{C}}(\theta) = \frac{\mathcal{Q}_{\mathbb{C}}(\theta)}{\sum_{\theta':\mathbb{C}} \mathcal{Q}_{\mathbb{C}}(\theta')}$ for any transaction $\theta : \mathbb{C}$.



(a) $\mathcal{P}(t_d) \cdot \mathcal{P}(t_e) \cdot \mathcal{P}(t_g) \cdot \mathcal{P}(t_b) = \frac{1}{4}$

(b) $\mathcal{P}(t_a) \cdot \mathcal{P}(t_e) \cdot \mathcal{P}(t_{bg}) = \frac{1}{8}$

(c) $\mathcal{P}(t_a) \cdot \mathcal{P}(t_e) \cdot \mathcal{P}(t_c) = \frac{1}{8}$

(d) $\mathcal{P}(t_a) \cdot \mathcal{P}(t_f) \cdot \mathcal{P}(t_b) = \frac{1}{4}$

Figure 12: Processes of the net $(\![\![N]\!]\!)$ (running example)

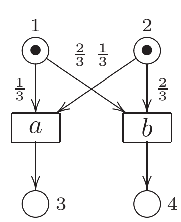


Figure 13: A free-choice net

Example 9. *Take the free-choice net in Fig. 13 and assume that decisions are local to each place. Thus, place* 1 *lends its token to* a *with probability* $p_1 = \frac{1}{3}$ *and to* b *with* $q_1 = \frac{2}{3}$*. Similarly, place* 2 *lends its token to* a *with probability* $p_2 = \frac{1}{3}$ *and to* b *with* $q_2 = \frac{2}{3}$*. Then one can set* $p_a = p_1 \cdot p_2 = \frac{1}{9}$ *and* $p_b = q_1 \cdot q_2 = \frac{4}{9}$*. However their sum is* $\frac{5}{9} \neq 1$*. This anomaly is due to the existence of deadlocked choices with nonzero probabilities which disappear in the process semantics of nets. To some extent, the probabilities assigned to* a *and* b *should be conditional w.r.t. the fact that the local choices performed at places* 1 *and* 2 *are compatible, i.e., all non compatible choices are disregarded. This means that we need to normalize the values of* p_a *and* p_b *over their sum. Of course, normalisation is possible only if there is at least one admissible alternative. In this simple example we get* $\mathcal{Q}(a) = \frac{1}{9}$*,* $\mathcal{Q}(b) = \frac{4}{9}$*,* $\mathcal{P}(a) = \frac{1}{9}/\frac{5}{9} = \frac{1}{5}$ *and* $\mathcal{P}(b) = \frac{4}{9}/\frac{5}{9} = \frac{4}{5}$*.*

Example 10. *Suppose that in our running example we are assigning uniform distributions to all arcs leaving a place. From simple calculation we have* $\mathcal{P}_{\mathbb{C}_1}(\theta_a) = \mathcal{P}_{\mathbb{C}_1}(\theta_d) = \frac{1}{2}$ *for the first cell,* $\mathcal{P}_{\mathbb{C}_2}(\theta_e) = \mathcal{P}_{\mathbb{C}_2}(\theta_f) = \frac{1}{2}$ *for the second cell,* $\mathcal{P}_{\mathbb{C}_3}(\theta_c) = \mathcal{P}_{\mathbb{C}_2}(\theta_{bg}) = \frac{1}{2}$ *for the third cell. The transactions of nested cells are uniquely defined and thus have all probability* 1*.* □

Given a firing sequence $t_1; \cdots ; t_n$ we can set $\mathcal{P}(t_1; \cdots ; t_n) = \prod_{i=1}^{n} \mathcal{P}(t_i)$. Hence firing sequences that differ in the order in which transitions are fired are assigned the same probability. Thanks to Theorem 3, we can consider maximal persistent processes and set $\mathcal{P}(O) = \prod_{t \in O} \mathcal{P}(t)$. In fact any maximal firing sequence in O includes all transitions of O. It follows from Theorem 3 that any maximal configuration has a corresponding maximal process (and viceversa) and since Abbes and Benveniste proved that the sum of the probabilities assigned to maximal configurations is 1, the same holds for maximal persistent processes.

Example 11. *Suppose that in our running example we are assigning uniform distributions to all choices available in an s-cell. Then the persistent process in Fig. 11 has probability:* $\mathcal{P}(O) = \mathcal{P}(t_d) \cdot \mathcal{P}(t_f) \cdot \mathcal{P}(t_3) \cdot \mathcal{P}(t_8) \cdot \mathcal{P}(t_b) \cdot \mathcal{P}(t'_8) = \frac{1}{2} \cdot \frac{1}{2} \cdot 1 \cdot 1 \cdot 1 \cdot 1 = \frac{1}{4}$*. There are other four maximal processes shown in Fig. 12 together with their probabilities. We note that the sum of all probabilities assigned to maximal processes is indeed* 1*.* □

7 Conclusion and Future Work

The problem of confusion is long-standing in Petri net theory. AB's branching cells provide a sort of interpreter (or scheduler) for executing PESs in the presence of confusion. We have extended their approach, by introducing s-cells and using nets with persistency, to compile their execution strategy in a statically defined, confusion-free, executable operational model. The advantage is that for p-nets the concurrency within a processes is consistent with the execution strategies, i.e., all linearizations of a persistent process are executable. Moreover, our definition is purely local (to s-cells), static and compositional, whereas AB's is dynamic and needs the entire PES be available.

Our construction is potentially complex. The construction of s-cells is to some extent hierarchical: given a s-cell $\mathbb{C}$ and its corresponding net $N_{\mathbb{C}}$ we recursively consider the nested s-cells arising from the decomposition of the net $N_{\mathbb{C}} \ominus p$, for any initial place $p \in N_{\mathbb{C}}$. This means that, in the worst case, when each s-cell is tightly connected, the number of nested s-cells can be exponential on the number of their initial places. However s-cells are typically much smaller than the whole net, so that even if the net grows very large it can be the case that the size of its s-cells is bound by some fixed k: in this case, the number of s-cells to be considered by our construction is still exponential on the fixed k, but linear w.r.t. the number of places of the net.

A limitation of our approach is that it applies to finite occurrence nets only (or, equivalently, to finite PESs). As a future work, we plan to extend the approach to deal with cycles and unfolding semantics. This requires some efforts and we conjecture it is feasible only if the net is safe and its behaviour has some form of regularity: the same s-cell can be executed several times in a computation but every instance is restarted without tokens left from previous ones. The causal AND/OR-dependencies share some similarities also with the work on connectors and Petri nets with boundaries [6] that we would like to formalize. We also want to investigate the connection between our s-cell structure and Bayesian networks, so to make forward and backward reasoning techniques available in our setting.

Acknowledgements The idea of defining structural branching cells emerged, albeit with a different objective, in collaboration with Lorenzo Galeotti after his Master thesis. We want to thank Glynn Winskel for pointing us to insightful references during the early preparation phase of this article.

References

[1] Samy Abbes and Albert Benveniste. Branching cells as local states for event structures and nets: Probabilistic applications. In Vladimiro Sassone, editor, *FOSSACS'05*, volume 3441, pages 95–109. Springer, 2005.

[2] Samy Abbes and Albert Benveniste. True-concurrency probabilistic models: Branching cells and distributed probabilities for event structures. *Inf. Comput.*, 204(2):231–274, 2006.

[3] Samy Abbes and Albert Benveniste. True-concurrency probabilistic models: Markov nets and a law of large numbers. *Theor. Comput. Sci.*, 390(2-3):129–170, 2008.

[4] Andrea Asperti and Nadia Busi. Mobile Petri nets. Technical Report Report UBLCS96-10, University of Bologna, May 1996.

[5] Andrea Asperti and Nadia Busi. Mobile Petri nets. *Mathematical Structures in Computer Science*, 19(6):1265–1278, 2009.

[6] Roberto Bruni, Hernán C. Melgratti, Ugo Montanari, and Pawel Sobocinski. Connector algebras for C/E and P/T nets' interactions. *Logical Methods in Computer Science*, 9(3), 2013.

[7] Federico Crazzolara and Glynn Winskel. Petri nets with persistence. *Electr. Notes Theor. Comput. Sci.*, 121:143–155, 2005.

[8] Joanne Bechta Dugan, Kishor S. Trivedi, Robert Geist, and Victor F. Nicola. Extended stochastic Petri nets: Applications and analysis. In Erol Gelenbe, editor, *Performance'84*, pages 507–519. North-Holland, 1984.

[9] Christian Eisentraut, Holger Hermanns, Joost-Pieter Katoen, and Lijun Zhang. A semantics for every GSPN. In José Manuel Colom and Jörg Desel, editors, *Petri Nets 2013*, volume 7927 of *Lecture Notes in Computer Science*, pages 90–109. Springer, 2013.

[10] Christian Eisentraut, Holger Hermanns, and Lijun Zhang. Concurrency and composition in a stochastic world. In Paul Gastin and François Laroussinie, editors, *CONCUR 2010*, volume 6269 of *Lecture Notes in Computer Science*, pages 21–39. Springer, 2010.

[11] Yrvann Emzivat, Benoît Delahaye, Didier Lime, and Olivier H. Roux. Probabilistic time Petri nets. In Fabrice Kordon and Daniel Moldt, editors, *ATPN'16*, volume 9698 of *Lecture Notes in Computer Science*, pages 261–280. Springer, 2016.

[12] Ursula Goltz and Wolfgang Reisig. The non-sequential behavior of Petri nets. *Information and Control*, 57(2/3):125–147, 1983.

[13] Benjamin Lucien Kaminski, Joost-Pieter Katoen, Christoph Matheja, and Federico Olmedo. Weakest precondition reasoning for expected run-times of probabilistic programs. In Peter Thiemann, editor, *ESOP 2016*, volume 9632 of *Lecture Notes in Computer Science*, pages 364–389. Springer, 2016.

[14] Joost-Pieter Katoen and Doron A. Peled. Taming confusion for modeling and implementing probabilistic concurrent systems. In Matthias Felleisen and Philippa Gardner, editors, *ESOP'13*, volume 7792 of *Lecture Notes in Computer Science*, pages 411–430. Springer, 2013.

[15] Marco Ajmone Marsan, Gianni Conte, and Gianfranco Balbo. A class of generalized stochastic Petri nets for the performance evaluation of multiprocessor systems. *ACM Trans. Comput. Syst.*, 2(2):93–122, 1984.

[16] Michael K. Molloy. Discrete time stochastic Petri nets. *IEEE Trans. Softw. Eng.*, 11(4):417–423, April 1985.

[17] Ugo Montanari and Francesca Rossi. Contextual nets. *Acta Inf.*, 32(6):545–596, 1995.

[18] Mogens Nielsen, Gordon D. Plotkin, and Glynn Winskel. Petri nets, event structures and domains. In Gilles Kahn, editor, *Proceedings of the International Symposium on Semantics of Concurrent Computation 1979*, volume 70 of *Lecture Notes in Computer Science*, pages 266–284. Springer, 1979.

[19] Mogens Nielsen, Gordon D. Plotkin, and Glynn Winskel. Petri nets, event structures and domains, part I. *Theor. Comput. Sci.*, 13:85–108, 1981.

[20] C.A. Petri. *Kommunikation mit Automaten.* PhD thesis, Institut für Instrumentelle Mathematik, Bonn, 1962.

[21] Wolfgang Reisig. *Understanding Petri Nets - Modeling Techniques, Analysis Methods, Case Studies.* Springer, 2013.

[22] Grzegorz Rozenberg and Joost Engelfriet. Elementary net systems. In Wolfgang Reisig and Grzegorz Rozenberg, editors, *Lectures on Petri Nets Part I: Basic Models, Advances in Petri Nets 1996*, volume 1491 of *Lecture Notes in Computer Science*, pages 12–121. Springer, 1998.

[23] Einar Smith. On the border of causality: Contact and confusion. *Theor. Comput. Sci.*, 153(1&2):245–270, 1996.

[24] Rob J. van Glabbeek, Ursula Goltz, and Jens-Wolfhard Schicke-Uffmann. On characterising distributability. *Logical Methods in Computer Science*, 9(3), 2013.

[25] Daniele Varacca, Hagen Völzer, and Glynn Winskel. Probabilistic event structures and domains. *Theor. Comput. Sci.*, 358(2-3):173–199, 2006.

[26] Ingo Wegener. *The complexity of Boolean functions.* Wiley-Teubner series in computer science. John Wiley & Sons, Inc., New York, NY, USA, 1987.

[27] Glynn Winskel. Event structures. In Wilfried Brauer, Wolfgang Reisig, and Grzegorz Rozenberg, editors, *Petri Nets: Central Models and Their Properties, Advances in Petri Nets 1986, Part II*, volume 255 of *Lecture Notes in Computer Science*, pages 325–392. Springer, 1987.

A Proofs of Main Results

This appendix presents the proof sketches of the main results. Note that we need some auxiliary lemmas that are not present in the main text of the paper. They are marked by the keyword "Aux" to avoid ambiguities.

For reader's convenience, full details are provided as complementary material in Appendix B.

A.1 Proofs of Results in Section 3

We start by showing that the encoding of a net into a dynamic net does not add computations. We show that each reachable marking b of the dynamic net can be associated with a reachable marking m of the original net, when disregarding negative information. We remark that in general the relation between such b and m is that $b \cap P \subseteq m$ and not necessarily $b \cap P = m$ (see, e.g., Lemma 4). This is because the transitions $t_{\theta,\mathbb{C}}$ generated by the encoding (T_{pos}) always consume the tokens in all minimal places of the branching cell $\mathbb{C}$. This choice is immaterial for the behaviour of the encoded net, as made explicit by the main results in the paper.

Lemma 4 (Aux.)**.** *Let $N = (P, T, F, m)$. If $[\![N]\!] \to^* (T, b)$ then $m \to^* m'$ and $b \cap P \subseteq m'$.*

Proof. The proof follows by induction on the length of the reduction $[\![N]\!] \to^n (T, b)$. $\square$

Corollary 5 (Aux.)**.** *If N is 1-safe then $[\![N]\!]$ is 1-∞-safe.*

Lemma 2. *Let $[\![N]\!] \in \text{DN}(P \cup \overline{\mathbf{P}})$. If $[\![N]\!] \to^* (T, b)$ then for all $t, t' \in T$ such that ${}^\bullet t \neq {}^\bullet t'$ and ${}^\bullet t \cap {}^\bullet t' \cap P \neq \emptyset$ it holds that there is $p \in P \cap ({}^\bullet t \cup {}^\bullet t')$ such that $\overline{\mathbf{p}} \in b$.*

Proof. The proof follows by induction on the length of the firing sequence $[\![N]\!] \xrightarrow{t_1 \cdots t_n} (T, b)$. $\square$

In what follows we write $p \prec q$ if $p \preceq q$ and $p \neq q$. The following auxiliary result provides some invariants about the configurations that can be reached by an encoded dynamic net.

Lemma 5 (Aux.)**.** *If $[\![N]\!] \to^* (T, b)$ then*

1. *$p \in b$ implies $\overline{\mathbf{p}} \notin b$;*
2. *if $\overline{\mathbf{p}} \in b$ and $p \preceq q$ then $q \notin b$;*
3. *if $p \preceq q$, $p \in b$ and $\overline{\mathbf{q}} \in b$ then there exists $r \prec q$ and $\overline{\mathbf{r}} \in b$; and*
4. *if $(T, b) \xrightarrow{t}$ and ${}^\bullet t = {}^\circ\mathbb{C}$ for some $\mathbb{C}$ then $(\mathbb{C}^\circ \cup \overline{\mathbb{C}^\circ}) \cap b = \emptyset$.*

Proof. The proof follows by induction on the length of the firing sequence $[\![N]\!] \xrightarrow{t_1 \cdots t_n} (T, b)$. $\square$

Lemma 1. *If $[\![N]\!] \to^* (T, b)$ and $\overline{\mathbf{p}} \in b$ then $(T, b) \to^* (T', b')$ implies that $p \notin b'$.*

Proof. If $\overline{\mathbf{p}} \in b$ then $\overline{\mathbf{p}} \in b'$ because $\overline{\mathbf{p}}$ is persistent. Moreover, $[\![N]\!] \to^* (T', b')$. By the contrapositive of Lemma 5(1), $p \notin b'$. $\square$

Theorem 1. *Let* $[\![N]\!] \in \text{DN}(P \cup \overline{\mathbf{P}})$. *If* $[\![N]\!] \to^* (T, b) \xrightarrow{t}$ *and* $(T, b) \xrightarrow{t'}$ *then either* $^\bullet t = {}^\bullet t'$ *or* $^\bullet t \cap {}^\bullet t' = \emptyset$.

Proof. By contradiction. Assume t, t' such that $(T, b) \xrightarrow{t}$, $(T, b) \xrightarrow{t'}$, $^\bullet t \neq {}^\bullet t'$, and $^\bullet t \cap {}^\bullet t' \neq \emptyset$. By construction of the encoding, it must be the case that $^\bullet t \subseteq P$ and $^\bullet t' \subseteq P$. Hence, $^\bullet t \cap {}^\bullet t' \cap P \neq \emptyset$. By Lemma 2 there exists $p \in P \cap ({}^\bullet t \cup {}^\bullet t')$ such that $\overline{\mathbf{p}} \in b$. By Lemma 1, $p \notin b$, which is in contradiction with the assumptions $(T, b) \xrightarrow{t}$ and $(T, b) \xrightarrow{t'}$. □

A.2 Proofs of Results in Section 4

We start by showing that reductions of a encoded net correspond to recursively stopped configurations of the event structure.

Lemma 6 (Aux.). *Let* $N = (P, T, F, m)$ *and* $\mathcal{E}$ *the event structure of* N. *If* $[\![N]\!] \xrightarrow{t_1 \cdots t_n} (T, b)$ *and* $v = \bigcup_{1 \leq i \leq n} \|t_i\|$, *then*

1. $b \cap P = {}^\circ\{e \mid e \in \mathcal{E}^v$ *and* $\lfloor e \rfloor = \{e\}\}$; *and*

2. *If* $(T, b) \xrightarrow{t}$ *then* $\|t\| \neq \emptyset$ *implies* $\|t\|$ *is a stopped configuration of* $\mathcal{E}^v$.

Proof. If follows by induction on the length of the reduction $[\![N]\!] \xrightarrow{t_1 \cdots t_n} (T, b)$. □

Lemma 7 (Aux.). *Let* $[\![N]\!] \in \text{DN}(P \cup \overline{\mathbf{P}})$. *If* $[\![N]\!] \to^* (T, b)$ *then there exists* (T', b') *such that* $(T, b) \Longrightarrow (T', b')$ *and*

1. $b' \cap P = b \cap P$;

2. *for all* p, q, *if* $\overline{\mathbf{p}} \in b$ *and* $p \preceq q$, *then* $\overline{\mathbf{q}} \in b'$;

3. *for all* $\mathbb{C} \in \text{BC}(N)$ *and* $\overline{\mathbf{Q}} \subseteq \overline{\mathbf{P}}$, *if* $\overline{\mathbf{Q}} \subseteq b'$ *then for all* $\mathbb{C}' \in \text{BC}(N_{\mathbb{C}} \ominus Q)$ *and* $\theta : \mathbb{C}'$ *there exists* $t \in T'$ *such that* $t = {}^\circ\mathbb{C}' \longrightarrow (\emptyset, \theta^\circ \cup \overline{\mathbb{C}'^\circ \setminus \theta^\circ})$.

Proof.
1. It follows straightforwardly by analysis of the applied rules. They are of the form $\overline{\mathbf{p}} \to (T', \overline{\mathbb{C}^\circ \setminus (N_{\mathbb{C}} \ominus p)^\circ})$, which does not consume nor produce tokens in regular places.

2. By induction on the length of the chain $p = p_0 \prec \ldots \prec p_n = q$ (this is a finite chain because N is a finite occurrence net). The inductive step follows by straightforward inspection of the shape of the transitions with negative premises.

3. By straightforward induction on the number n of elements in $\overline{\mathbf{Q}}$, i.e., $n = |\overline{\mathbf{Q}}|$.

□

Lemma 8 (Aux). *Let* $N = (P, T, F, m)$ *and* $\mathcal{E}$ *the event structure of* N. *If* v *is recursively stopped configuration and* $v = \bigcup_{1 \leq i \leq n} v_i$ *is a valid decomposition, then*

1. $[\![N]\!] \xRightarrow{t_1 \cdots t_n} (T, b)$ *and* $v = \bigcup_{1 \leq i \leq n} \|t_i\|$*;*
2. $b \cap P = {}^\circ\{e \mid e \in \mathcal{E}^v$ *and* $\lfloor e \rfloor = \{e\}\}$*;*
3. *If* v' *is a stopped configuration of* $\mathcal{E}^v$*, then there exists* $t \in T$ *s.t.* $(T, b) \xrightarrow{t}$ *and* $\|t\| = v'$*;*
4. *For all* $e \in \mathcal{E}$*, if* $e \notin (\mathcal{E}^v \cup v)$ *implies* $\overline{{}^\bullet e} \cap b = 0$*.*

Proof. If follows by induction on the length n of the decomposition $v = \bigcup_{1 \leq i \leq n} v_i$. □

Lemma 3. *Let* N *be an occurrence net.*

1. *If* $[\![N]\!] \xRightarrow{t_0 \cdots t_n}$*, then* $v = \bigcup_{1 \leq i \leq n} \|t_i\|$ *is recursively-stopped in* $\mathcal{E}_N$ *and* $(\|t_i\|)_{1 \leq i \leq n}$ *is a valid decomposition of* v*.*
2. *If* v *is recursively-stopped in* $\mathcal{E}_N$*, then for any valid decomposition* $(v_i)_{1 \leq i \leq n}$ *there exists* $[\![N]\!] \xRightarrow{t_0 \cdots t_n}$ *such that* $\|t_i\| = v_i$*.*

Proof. 1). It follows from Lemma 6. 2). It follows from Lemma 8(1). □

Theorem 2. *Let* N *be an occurrence net.*

1. *If* $(\![[\![N]\!]]\!) \xRightarrow{t_1 \cdots t_n}$*, then* $v = \bigcup_{1 \leq i \leq n} \|t_i\|$ *is recursively-stopped in* $\mathcal{E}_N$ *and* $(\|t_i\|)_{1 \leq i \leq n}$ *is a valid decomposition of* v*.*
2. *If* v *is recursively-stopped in* $\mathcal{E}_N$*, then for any valid decomposition* $(v_i)_{1 \leq i \leq n}$ *there exists* $(\![[\![N]\!]]\!) \xRightarrow{t_1 \cdots t_n}$ *such that* $\|t_i\| = v_i$*.*

Proof. It follows from Lemma 3 and Proposition 1. □

A.3 Proofs of Results in Section 5

Theorem 3. *Let* $\sigma = t_1; \cdots; t_n$ *with* $n \geq 0$ *be a, possibly empty, firing sequence of a persistent process, and* t *a transition not in* σ*. The following conditions are all equivalent: (i)* t *is enabled after* σ*; (ii) there is a collection of causes of* t *which appears in* σ*; (iii)* $\bigwedge_{i=1}^{n} t_i$ *implies* $\Phi(t)$*.*

Proof. **ii)** $\Leftrightarrow$ **(iii):** We have that $\bigwedge_{i=1}^{n} t_i$ implies $\Phi(t)$ iff there is a prime implicant $\bigwedge_{j=1}^{m} t_{i_j}$ of $\Phi(t)$ that is implied by $\bigwedge_{i=1}^{n} t_i$. This is the case iff the collection of causes $\{t_{i_1}, ..., t_{i_m}\}$ appears in σ.

(i) $\Rightarrow$ **(iii):** The proof is by induction on the length n of the sequence.

For the base case, if $n = 0$ it means that t is enabled in the initial marking, i.e., that its pre-set only contains initial places of the process and thus $\Phi(t) = true$.

For the inductive case, assume the property holds for any shorter sequence $t_1; \cdots; t_k$ with $0 \leq k < n + 1$ and let us prove that it holds for $\sigma =$

$t_1; \cdots; t_{n+1}$. Let b_0 the initial bag of the process. As t is enabled after σ we have $b_0 \xrightarrow{\sigma} b \xrightarrow{t}$ for some bag b. Since t is enabled in b, we have ${}^\bullet t \subseteq b$, i.e., for any $s \in {}^\bullet t$ we have $b(s) \in \{1, \infty\}$ (by definition of p-net, ${}^\bullet t$ is not empty). We need to prove that $\Phi(t) = \wedge_{s \in {}^\bullet t} \Phi(s)$ is implied by $\bigwedge_{i=1}^{n+1} t_i$, i.e., that for any $s \in {}^\bullet t$ the formula $\Phi(s)$ is implied by $\bigwedge_{i=1}^{n+1} t_i$. Take a generic $s \in {}^\bullet t$. Either ${}^\bullet s = \emptyset$, in which case s is initial and $\Phi(s) = \mathit{true}$, or ${}^\bullet s \neq \emptyset$ and $\Phi(s) = \bigvee_{t' \in {}^\bullet s}(t' \wedge \Phi(t'))$. Since $b(s) \in \{1, \infty\}$, there must exist an index $j \in [1, n+1]$ such that $t_j \in {}^\bullet s$. Take $t' = t_j$. Since σ is a firing sequence, the transition t_j is enabled after $\sigma' = t_1; \cdots; t_{j-1}$. As $k = j - 1 < n + 1$, by inductive hypothesis $\Phi(t_j)$ is implied by $\bigwedge_{i=1}^{j-1} t_i$ and thus also by $\bigwedge_{i=1}^{n+1} t_i$. Since $\bigwedge_{i=1}^{n+1} t_i$ clearly implies t_j we have that $\bigwedge_{i=1}^{n+1} t_i$ implies $\Phi(s) = t_j \wedge \Phi(t_j)$.

(iii) $\Rightarrow$ **(i):** Suppose $\bigwedge_{i=1}^{n} t_i$ implies $\Phi(t) = \bigwedge_{s \in {}^\bullet t} \Phi(s)$. If for all $s \in {}^\bullet t$ we have ${}^\bullet s = \emptyset$, then t is enabled in the initial marking and as the process is deterministic no transition can steal tokens from ${}^\bullet t$ and t remains enabled after the firing of any $\sigma = t_1; \cdots; t_n$. Otherwise, $\Phi(t) = \bigwedge_{s \in {}^\bullet t} \bigvee_{t' \in {}^\bullet s \neq \emptyset}(t' \wedge \Phi(t'))$. Thus, for any $s \in {}^\bullet t$ with ${}^\bullet s \neq \emptyset$ there exists some $t' \in {}^\bullet s$ such that $\bigwedge_{i=1}^{n} t_i$ implies $t' \wedge \Phi(t')$. Since $\bigwedge_{i=1}^{n} t_i$ implies t' then there exists some index $k \in [1, n]$ such that $t' = t_k$ and s becomes marked during the firing of σ. As the process is deterministic, no transition can steal tokens from s. Since all the places in the pre-set of t becomes marked during the firing of σ, then t is enabled after σ.

$\square$

B Full Proofs

B.1 Detailed proofs of results in Section 2.4

Proposition 1. *Let $N = (T, b) \in \textsc{dn}(\mathbb{S})$. Then,*

1. *$N \xrightarrow{t} N'$ implies $(\!|N|\!) \xrightarrow{t} (\!|N'|\!)$;*

2. *Moreover, $(\!|N|\!) \xrightarrow{t} N'$ implies there exists N'' such that $N \xrightarrow{t} N''$ and $N'' = (\!|N'|\!)$.*

Proof. We start by showing that $N \xrightarrow{t} N'$ implies $(\!|N|\!) \xrightarrow{t} (\!|N'|\!)$. If $N \xrightarrow{t} N'$ then $t = S \to (T', b') \in T$, $S \subseteq b$ and $N' = (T \cup T', (b \setminus S) \cup b')$. By definition of $(\!|\_|\!)$, it holds that $(\!|N|\!) = (\mathbb{S} \cup \mathbf{P}_{\mathbb{T}(N)}, \mathbb{T}(N), F, b \cup b_T)$ where $\mathbf{P}_{\mathbb{T}(N)} = \{\mathbf{p}_{t'} \mid t' \in \mathbb{T}(N)\}$. Note that

$$
\begin{array}{rll}
\mathbb{T}(N) & = T \cup \bigcup_{t' \in T} \mathbb{T}(t'^{\bullet}) & \text{by def. of } \mathbb{T}(\_) \\
& = T \cup \bigcup_{t' \in T} \mathbb{T}(t'^{\bullet}) \cup \mathbb{T}(t^{\bullet}) & t \in T \\
& = T \cup \bigcup_{t' \in T} \mathbb{T}(t'^{\bullet}) \cup T' \cup \bigcup_{t' \in T'} \mathbb{T}(t'^{\bullet}) & \text{by def. of } \mathbb{T}(\_) \\
& = (T \cup T') \cup \bigcup_{t' \in T \cup T'} \mathbb{T}(t'^{\bullet}) & \text{by assoc. and comm. of } \cup \\
& = \mathbb{T}(N') & \text{by def. of } \mathbb{T}(\_)
\end{array}
$$

Hence, $(\!|N'|\!) = (\mathbb{S} \cup \mathbf{P}_{\mathbb{T}(N)}, \mathbb{T}(N), F, (b \setminus S) \cup b' \cup b_{T \cup T'})$. By the definition of $(\!|\_|\!)$, $t \in \mathbb{T}(N)$ and F is such that $t : S \cup \{\mathbf{p}_t\} \to b' \cup b_{T'}$. Hence, t is enabled in $b \cup b_T$ because $S \subseteq b$ and $\mathbf{p}_t \in b_T$. Consequently, $b \cup b_T \xrightarrow{t} ((b \cup b_T) \setminus (S \cup \{\mathbf{p}_t\})) \cup b' \cup b_{T'} = (b \setminus S) \cup b_T \cup b' \cup b_{T'} = (b \setminus S) \cup b' \cup b_{T \cup T'}$.

The proof for $(\!|N|\!) \xrightarrow{t} N'$ implies there exists N'' such that $N \xrightarrow{t} N''$ and $N'' = (\!|N'|\!)$ follows by analogous arguments. □

B.2 Detailed proofs of results in Section 3

Lemma 4. *Let $N = (P, T, F, m)$. If $[\![N]\!] \to^* (T, b)$ then $m \to^* m'$ and $b \cap P \subseteq m'$.*

Proof. The proof follows by induction on the length of the reduction $[\![N]\!] \to^n (T, b)$.

- **Base case (n=0)**. It follows immediately because $b = m$.

- **Inductive case (n = k+1)**. Then, $[\![N]\!] \to^k (T', b') \xrightarrow{t} (T, b)$. By inductive hypothesis, $m \to^* m''$ and $b' \cap P \subseteq m''$. We now proceed by case analysis on the shape of t.

  – $t = {}^{\circ}\mathbb{C} \to (\emptyset, \theta^{\circ} \cup \overline{\mathbb{C}^{\circ} \setminus \theta^{\circ}})$. Then, ${}^{\circ}\mathbb{C} \subseteq b'$, $T = T'$ and $b = (b' \setminus {}^{\circ}\mathbb{C}) \cup \theta^{\circ} \cup \overline{\mathbb{C}^{\circ} \setminus \theta^{\circ}}$. Since ${}^{\circ}\mathbb{C} \subseteq P$, we have ${}^{\circ}\mathbb{C} \subseteq m''$. Moreover, $\theta : \mathbb{C}$ implies ${}^{\circ}\theta \subseteq {}^{\circ}\mathbb{C} \subseteq m''$. Since θ is a deterministic process, $m'' \to^* (m'' \setminus {}^{\circ}\theta) \cup \theta^{\circ}$. Then, take $m' = (m'' \setminus {}^{\circ}\theta) \cup \theta^{\circ}$.
  Note that $b \cap P = ((b' \cap P) \setminus {}^{\circ}\mathbb{C}) \cup \theta^{\circ}$. We use $(b' \cap P) \subseteq m''$ and ${}^{\circ}\theta \subseteq {}^{\circ}\mathbb{C}$ to conclude that $b \cap P \subseteq m'$.

– $t = \overline{\mathbf{p}} \rightarrow (T', \overline{\mathbb{C}^\circ \setminus (N_{\mathbb{C}} \ominus p)^\circ})$. It follows immediately because $b' \cap P = b \cap P$.

$\square$

Lemma 2. *Let $[\![N]\!] \in \text{DN}(P \cup \overline{\mathbf{P}})$. If $[\![N]\!] \rightarrow^* (T, b)$ then for all $t, t' \in T$ such that ${}^\bullet t \neq {}^\bullet t'$ and ${}^\bullet t \cap {}^\bullet t' \cap P \neq \emptyset$ it holds that there is $p \in P \cap ({}^\bullet t \cup {}^\bullet t')$ such that $\overline{\mathbf{p}} \in b$.*

Proof. The proof follows by induction on the length of the firing sequence $[\![N]\!] \xrightarrow{t_1 \cdots t_n} (T, b)$.

- **Base Case** $n = 0$. It holds trivially because any pair of different transitions in T have either the same preset (i.e., if they are taken from T_{pos} and originate from the same s-cell) or disjoint presets (i.e., if they are taken both from T_{pos} but originate from different s-cells, or both from T_{neg}, or one from T_{neg} and the other from T_{pos}).

- **Inductive step** $n = k + 1$. Hence, $[\![N]\!] \xrightarrow{t_1 \cdots t_k} (T', b') \xrightarrow{t_{k+1}} (T, b)$. By inductive hypothesis, for all $t, t' \in T'$ such that and ${}^\bullet t \cap {}^\bullet t' \cap P \neq \emptyset$, it holds that there is $p \in P \cap ({}^\bullet t \cup {}^\bullet t')$ such that $\overline{\mathbf{p}} \in b'$. Then, we proceed by case analysis on t_{k+1}.

  – $t_{k+1} = {}^\circ\mathbb{C} \rightarrow (\emptyset, \theta^\circ \cup \overline{\mathbb{C}^\circ \setminus \theta^\circ})$. It holds trivially because $T = T'$ and $b' \cap \overline{\mathbf{P}} \subseteq b$.

  – $t_{k+1} = \overline{\mathbf{p}} \rightarrow (T'', \overline{\mathbb{C}^\circ \setminus (N_{\mathbb{C}} \ominus p)^\circ})$ for some $\mathbb{C}$, $p \in {}^\circ\mathbb{C}$, and $(T'', \emptyset) = [\![N_{\mathbb{C}} \ominus p]\!]$. Then $T = T' \cup T''$. By the definition of $[\![\_]\!]$, we have that for all $t, t' \in T''$ either (i) ${}^\bullet t = {}^\bullet t'$ or (ii) ${}^\bullet t \cap {}^\bullet t' = \emptyset$ (reasoning analogously to the Base Case). It remains to consider the cases in which t and t' are taken one from T' and the other from T''. W.l.o.g., we consider $t \in T''$ and $t' \in T'$ and proceed as follows. Note that, by construction of $[\![N]\!]$, ${}^\bullet t \subseteq \overline{\mathbf{P}}$ implies $|{}^\bullet t| = 1$ for any t. Hence, the only possibility is ${}^\bullet t = {}^\circ\mathbb{C}_1$ with $\mathbb{C}_1 \in \text{BC}(N_{\mathbb{C}} \ominus p)$ and ${}^\bullet t' = {}^\circ\mathbb{C}_2$ with $\mathbb{C}_1 \neq \mathbb{C}_2$ and ${}^\circ\mathbb{C}_1 \cap {}^\circ\mathbb{C}_2 \neq \emptyset$. Note that ${}^\bullet t \cup {}^\bullet t' \subseteq P$. We proceed by contradiction and assume $\overline{({}^\bullet t \cup {}^\bullet t')} \cap b = \emptyset$. There must exist a s-cell $\mathbb{C}_3$ such that $\mathbb{C}_1 \cup \mathbb{C}_2 \subseteq \mathbb{C}_3$ (because $\mathbb{C}_1$ and $\mathbb{C}_2$ are closed under immediate conflict and their union introduces immediate conflict between the transitions consuming from the shared places in ${}^\circ\mathbb{C}_1 \cap {}^\circ\mathbb{C}_2$). If $\mathbb{C}_2 = \mathbb{C}_3$, then $\mathbb{C}_1 \subset \mathbb{C}_2$ and hence $p \in {}^\circ\mathbb{C}_2$ and $p \in {}^\bullet t'$, which contradicts $\overline{({}^\bullet t \cup {}^\bullet t')} \cap b = \emptyset$ because t_{k+1} enabled at b implies $\overline{\mathbf{p}} \in b$. Otherwise, $\mathbb{C}_2 \subset \mathbb{C}_3$. Consequently, there exists (at least) a transition $t'' \in T'$ such that ${}^\bullet t'' = {}^\circ\mathbb{C}_3$ and ${}^\bullet t' \neq {}^\bullet t'''$. Since $t' \in T'$ and $t'' \in T'$, we can use inductive hypothesis to conclude that $\overline{({}^\bullet t' \cup {}^\bullet t'')} \cap b \neq \emptyset$. The proof is completed by noting that this is in contradiction with the assumption $\overline{({}^\bullet t \cup {}^\bullet t')} \cap b \neq \emptyset$ because ${}^\bullet t'' = {}^\bullet t \cup {}^\bullet t'$.

$\square$

Lemma 5. *If* $[\![N]\!] \to^* (T, b)$ *then*

1. $p \in b$ *implies* $\overline{\mathbf{p}} \notin b$*;*
2. *if* $\overline{\mathbf{p}} \in b$ *and* $p \preceq q$ *then* $q \notin b$*;*
3. *if* $p \preceq q$*,* $p \in b$ *and* $\overline{\mathbf{q}} \in b$ *then there exists* $r \prec q$ *and* $\overline{\mathbf{r}} \in b$*; and*
4. *if* $(T, b) \xrightarrow{t}$ *and* ${}^\bullet t = {}^\circ\mathbb{C}$ *for some* $\mathbb{C}$ *then* $(\mathbb{C}^\circ \cup \overline{\mathbb{C}^\circ}) \cap b = \emptyset$*.*

Proof. The proof follows by induction on the length of the firing sequence $[\![N]\!] \xrightarrow{t_1 \cdots t_n} (T, b)$.

- **Base Case** $n = 0$. Hence $(T, b) = [\![N]\!]$.
  1. It follows from $b \subseteq P$.
  2. Since $b \subseteq P$ there is no $\overline{\mathbf{p}} \in b$.
  3. Since $b \subseteq P$ there is no $\overline{\mathbf{q}} \in b$.
  4. It follows from the fact that N is an occurrence net, $b \subseteq {}^\circ N$, and hence there does not exist any $\mathbb{C} \in \textsc{bc}(N)$ such that $\mathbb{C}^\circ \cap b \neq \emptyset$.

- **Inductive step** $n = k + 1$. Hence, $[\![N]\!] \xrightarrow{t_1 \cdots t_k} (T', b') \xrightarrow{t_{k+1}} (T, b)$. By inductive hypothesis, (1) $p \in b'$ implies $\overline{\mathbf{p}} \notin b'$; (2) if $\overline{\mathbf{p}} \in b'$ and $p \preceq q$ then $q \notin b'$; (3) if $p \preceq q$, $p \in b'$ and $\overline{\mathbf{q}} \in b'$ then there exists $r \prec q$ and $\overline{\mathbf{r}} \in b'$; and (4) if $(T', b') \xrightarrow{t}$ and ${}^\bullet t = {}^\circ\mathbb{C}$ for some $\mathbb{C}$ then $(\mathbb{C}^\circ \cup \overline{\mathbb{C}^\circ}) \cap b' = \emptyset$. We now proceed by case analysis on t_{k+1}.
  - $t_{k+1} = {}^\circ\mathbb{C}_{k+1} \to (\emptyset, \theta^\circ \cup \overline{\mathbb{C}^\circ_{k+1} \setminus \theta^\circ})$ for some s-cell $\mathbb{C}_{k+1}$ and transaction $\theta : \mathbb{C}_{k+1}$. Hence, $b = (b' \setminus {}^\circ\mathbb{C}_{k+1}) \cup (\theta^\circ \cup \overline{\mathbb{C}^\circ_{k+1} \setminus \theta^\circ})$.
    1. We proceed by contradiction. Assume that there exists p such that $p \in b$ and $\overline{\mathbf{p}} \in b$. Since $p \in b$ we have that either $p \in b' \setminus {}^\circ\mathbb{C}_{k+1}$ or $p \in \theta^\circ$. First, assume $p \in b' \setminus {}^\circ\mathbb{C}_{k+1}$. By inductive hypothesis (1), $\overline{\mathbf{p}} \notin b'$ and, hence, $\overline{\mathbf{p}} \notin b' \setminus {}^\circ\mathbb{C}_{k+1}$. Therefore, it should be the case that $\overline{\mathbf{p}} \in (\theta^\circ \cup \overline{\mathbb{C}^\circ_{k+1} \setminus \theta^\circ})$. Hence, $\overline{\mathbf{p}} \in \overline{\mathbb{C}^\circ_{k+1}}$ and $p \in \mathbb{C}^\circ_{k+1}$. Since t_{k+1} is enabled at (T', b'), we can use inductive hypothesis (4) on t_{k+1} to conclude $(\mathbb{C}^\circ_{k+1} \cup \overline{\mathbb{C}^\circ_{k+1}}) \cap b' = \emptyset$. Consequently, $p \in \mathbb{C}^\circ_{k+1}$ implies $p \notin b'$. But this is in contradiction with the assumption that $p \in b' \setminus {}^\circ\mathbb{C}_{k+1}$. Assume instead $p \in \theta^\circ$. Then $\overline{\mathbf{p}} \notin \overline{\mathbb{C}^\circ_{k+1} \setminus \theta^\circ}$. Hence, it should be the case that $\overline{\mathbf{p}} \in b' \setminus {}^\circ\mathbb{C}_{k+1}$. But this is also in contradiction with the the hypothesis (4) $(\mathbb{C}^\circ_{k+1} \cup \overline{\mathbb{C}^\circ_{k+1}}) \cap b' = \emptyset$.
    2. We proceed by contradiction. Assume there exist p and q such that $\overline{\mathbf{p}} \in b$, $p \preceq q$ and $q \in b$. Assume $\overline{\mathbf{p}} \in b'$. By inductive hypothesis (2), for all q s.t. $p \preceq q$ it holds that $q \notin b' \setminus {}^\circ\mathbb{C}_{k+1}$. Moreover, if $q \in (\theta^\circ \cup \overline{\mathbb{C}^\circ_{k+1} \setminus \theta^\circ})$ implies $p' \preceq q$ for all $p' \in {}^\circ\mathbb{C}_{k+1}$ by definition of branching cells. Since t is enabled at b', ${}^\circ\mathbb{C}_{k+1} \subseteq$

b' and hence $p \in b'$, but this is in contradiction with inductive hypothesis (1), i.e., $\overline{\mathbf{p}} \in b'$ implies $p \notin b'$. Assume instead $\overline{\mathbf{p}} \in \theta^\circ \cup \overline{\mathbb{C}^\circ_{k+1} \setminus \theta^\circ}$. Hence, $p \in \mathbb{C}^\circ_{k+1} \setminus \theta^\circ$ and $p \in \mathbb{C}^\circ_{k+1}$. Suppose there is $q \in b'$ and $p \preceq q$. Note that $p' \preceq p$ for all $p' \in {}^\circ\mathbb{C}_{k+1}$ by definition of branching cells. By transitivity of $\preceq$, $p' \preceq q$ for all $p' \in {}^\circ\mathbb{C}_{k+1}$. Since t is enabled at b', ${}^\circ\mathbb{C}_{k+1} \subseteq b'$. By using Lemma 4, we can conclude that $q \notin b'$ for all q s.t. $p \preceq q$, which contradicts the hypothesis $q \in b'$ and $p \preceq q$. Assume instead $q \in \theta^\circ \cup \overline{\mathbb{C}^\circ_{k+1} \setminus \theta^\circ}$. Hence, $q \in \theta^\circ$. Hence, $p \neq q$. Moreover, $p \in \mathbb{C}^\circ_{k+1}$ and $q \in \mathbb{C}^\circ_{k+1}$ contradict the hypothesis $p \preceq q$.

3. If $\overline{\mathbf{q}} \in b'$ the proof follows by inductive hypothesis and by noting that $p \in \theta^\circ$ and $p \preceq q$ imply there exists $r \in {}^\circ\mathbb{C}$ and $r \preceq q$ (by transitivity of $\preceq$). If $\overline{\mathbf{q}} \in \theta^\circ \cup \overline{\mathbb{C}^\circ_{k+1} \setminus \theta^\circ}$, follows by contradiction because $p \preceq q$ and $p \in b'$ implies $p \in {}^\circ\mathbb{C}$ by Lemma 4. Therefore, there does not exist p such that $p \preceq q$ and $p \in b$.

4. Let $t \in T$ such that ${}^\bullet t = {}^\circ\mathbb{C} \subseteq b$ for some $\mathbb{C}$. Since t is enabled at b and $[\![N]\!]$ is 1-∞-safe by Corollary 5, then $\mathbb{C}_{k+1} \cap \mathbb{C} = \emptyset$. If t is enabled at $(b' \setminus {}^\circ\mathbb{C}_{k+1})$ then t is enabled at b'. By inductive hypothesis (2), we conclude that $(\mathbb{C}^\circ \cup \overline{\mathbb{C}^\circ}) \cap (b' \setminus {}^\circ\mathbb{C}_{k+1}) = \emptyset$. If t is not enabled at (T', b'), then it holds that for $x \in \mathbb{C}$ exists $y \in (\mathbb{C}^\circ_{k+1} \cup \overline{\mathbb{C}^\circ_{k+1}})$ such that $y \preceq x$. By inductive hypothesis $(\mathbb{C}^\circ_{k+1} \cup \overline{\mathbb{C}^\circ_{k+1}}) \cap b' = \emptyset$, hence $(\mathbb{C}^\circ \cup \overline{\mathbb{C}^\circ}) \cap b' = \emptyset$. Therefore, $(\mathbb{C}^\circ \cup \overline{\mathbb{C}^\circ}) \cap b = \emptyset$.

– $t_{k+1} = \overline{\mathbf{r}} \to (T'', \overline{\mathbb{C}^\circ_{k+1} \setminus (N_{\mathbb{C}_{k+1}} \ominus r)^\circ})$ for some s-cell $\mathbb{C}_{k+1}$ and place $r \in {}^\circ\mathbb{C}_{k+1}$. Then, $T = T' \cup T''$ with $[\![N_{\mathbb{C}_{k+1}} \ominus r]\!] = (T'', \_)$ and $b = b' \cup \overline{\mathbb{C}^\circ_{k+1} \setminus (N_{\mathbb{C}_{k+1}} \ominus r)^\circ}$.

1. We proceed by contradiction. Assume that there exists p such that $p \in b$ and $\overline{\mathbf{p}} \in b$. Note that $p \in b$ implies $p \in b'$. By inductive hypothesis (1), $\overline{\mathbf{p}} \notin b'$. Therefore, it should be the case that $\overline{\mathbf{p}} \in \overline{\mathbb{C}^\circ_{k+1} \setminus (N_{\mathbb{C}_{k+1}} \ominus r)^\circ}$. Consequently $p \in \mathbb{C}^\circ_{k+1}$ and $p \notin (N_{\mathbb{C}_{k+1}} \ominus r)^\circ$. Hence, $r \preceq p$. Since t is enabled at b', $\overline{\mathbf{r}} \in b'$. By inductive hypothesis (2), $p \notin b'$ which contradicts the hypothesis $p \notin b$.

2. We proceed by contradiction. Assume there exist p and q such that $\overline{\mathbf{p}} \in b$, $p \preceq q$ and $q \in b$. Note that $q \in b$ implies $q \in b'$. Assume $\overline{\mathbf{p}} \in b'$. By inductive hypothesis, for all q s.t. $p \preceq q$ then $q \notin b'$ and, hence it is in contradiction with assumption $q \in b$. Assume instead $\overline{\mathbf{p}} \in \overline{\mathbb{C}^\circ_{k+1} \setminus (N_{\mathbb{C}_{k+1}} \ominus r)^\circ}$. As before, we conclude that $r \preceq p$. By transitivity of $\preceq$, we have $r \preceq q$. By inductive hypothesis (2), $q \notin b'$, which is in contradiction with assumption $q \in b$.

3. For $\overline{\mathbf{q}} \in b$, it follows immediately by inductive hypothesis. For $\overline{\mathbf{q}} \in \overline{\mathbb{C}^\circ_{k+1} \setminus (N_{\mathbb{C}_{k+1}} \ominus r)^\circ}$, it follows straightforwardly because $r \preceq q$ and $\overline{\mathbf{r}} \in b$.

4. Assume ${}^\bullet t = {}^\circ\mathbb{C} \subseteq b$ for some $\mathbb{C}$. Hence, ${}^\bullet t = {}^\circ\mathbb{C} \subseteq b'$. There are two cases:
   * Suppose $t \in T'$. By inductive hypothesis (4), $(\mathbb{C}^\circ \cup \overline{\mathbb{C}^\circ}) \cap b' = \emptyset$. We show that $(\mathbb{C}^\circ \cup \overline{\mathbb{C}^\circ}) \cap \overline{\mathbb{C}^\circ_{k+1} \setminus (N_{\mathbb{C}_{k+1}} \ominus r)^\circ} = \emptyset$ also holds. It is enough to show that $\overline{\mathbb{C}^\circ} \cap \overline{\mathbb{C}^\circ_{k+1} \setminus (N_{\mathbb{C}_{k+1}} \ominus r)^\circ} = \emptyset$. We proceed by contradiction and assume there exists q such that $q \in \mathbb{C}^\circ$ and $q \in (\mathbb{C}^\circ_{k+1} \setminus (N_{\mathbb{C}_{k+1}} \ominus r)^\circ)$. Because $q \in (\mathbb{C}^\circ_{k+1} \setminus (N_{\mathbb{C}_{k+1}} \ominus r)^\circ)$, $r \preceq q$. Since $q \in \mathbb{C}^\circ$, $r \in {}^\circ\mathbb{C}$ (because $\mathbb{C}$ is closed under causality). Hence $r \in b'$ because t is enabled at b'. By the contrapositive of inductive hypothesis (1), $\overline{\mathbf{r}} \notin b'$, but this is in contradiction with the hypothesis that t_{k+1} is enabled at b'.
   * Suppose $t \in T''$. Then, ${}^\bullet t \cap {}^\circ(N_{\mathbb{C}_{k+1}} \ominus r) = \emptyset$. for some $\mathbb{C}$. Hence, for all $q \in \mathbb{C}^\circ$ there exists $s \in {}^\circ\mathbb{C}$ s.t. $s \preceq q$. Since t is enabled at b, ${}^\circ\mathbb{C} \subseteq b$ holds. By Lemma 4, $\mathbb{C}^\circ \cap b = \emptyset$. We show by contradiction that $\overline{\mathbb{C}^\circ} \cap b = \emptyset$ does not hold either. Assume that there exists $\overline{\mathbf{q}} \in \overline{\mathbb{C}^\circ}$ and $\overline{\mathbf{q}} \in b$. Since there exists $s \in \mathbb{C} \subseteq b$ and $s \preceq q$, we can use inductive hypothesis (3) to conclude that there exist $\overline{\mathbf{s'}} \subseteq b$ and $s' \preceq q$ and $s' \in {}^\circ\mathbb{C}$. By the inductive hypothesis (1) $s' \notin b$, and this is in contradiction with the assumption of t enabled at b.

$\square$

B.3 Detailed proofs of results in Section 4

This section is devoted to prove the main results in Section 4. We start by providing some auxiliary results.

Lemma 6. *Let $N = (P, T, F, m)$ and $\mathcal{E}$ the event structure of N. If $[\![N]\!] \xrightarrow{t_1 \cdots t_n} (T, b)$ and $v = \bigcup_{1 \leq i \leq n} \|t_i\|$, then*

1. *$b \cap P = {}^\circ\{e \mid e \in \mathcal{E}^v$ and $\lfloor e \rfloor = \{e\}\}$; and*
2. *If $(T, b) \xrightarrow{t}$ then $\|t\| \neq \emptyset$ implies $\|t\|$ is a stopped configuration of $\mathcal{E}^v$.*

Proof. If follows by induction on the length of the reduction $[\![N]\!] \xrightarrow{t_1 \cdots t_n} (T, b)$.

- **Base case (n=0)**. Then, $v = \emptyset$ and $\mathcal{E}^v = \mathcal{E}$. Moreover, $b = m$.
  1. It is immediate to notice that m corresponds to the preset of all minimal events of $\mathcal{E}$.
  2. Since t is enabled, ${}^\bullet t \subseteq m$. Hence, ${}^\bullet t = {}^\circ\mathbb{C}$ with $\mathbb{C} \in \textsc{bc}(N)$. Therefore, $\mathbb{C}$ corresponds to a branching cell of $\mathcal{E}$. By the definition of $[\![\_]\!]$, t is associated with some $\theta : \mathbb{C}$, which is a maximal, conflict-free set of transitions in $\mathbb{C}$. Hence, $\|t\|$ is a stopped configuration of $\mathcal{E}$.

- **Inductive case (n = k+1)**. Then, $[\![N]\!] \xrightarrow{t_1\cdots t_k} (T_k, b_k) \xrightarrow{t_{k+1}} (T, b)$. By inductive hypothesis, letting $v_k = \bigcup_{1\leq i\leq k}\|t_i\|$, we assume (1) $b_k \cap P = {}^\circ\{e \mid e \in \mathcal{E}^{v_k}$ and $\lfloor e\rfloor = \{e\}\}$, and (2) If $(T_k, b_k) \xrightarrow{t}$ then $\|t\| \neq \emptyset$ implies $\|t\|$ is a stopped configuration of $\mathcal{E}^{v_k}$.

  We now proceed by case analysis on the shape of the applied rule:

  - $t_{k+1} = {}^\circ\mathbb{C} \to (\emptyset, \theta^\circ \cup \overline{\mathbb{C}^\circ \setminus \theta^\circ})$. Hence, $v = v_k \cup \|\theta\|$ and $b \cap P = (b_k \cap P \setminus {}^\circ\mathbb{C}) \cup \theta^\circ$.
    1. Then:

       $$\begin{array}{rl}\{e \mid e \in \mathcal{E}^v \wedge \lfloor e\rfloor = \{e\}\} = & \{e \mid e \in \mathcal{E}^{v_k} \wedge e \notin \mathbb{C} \wedge \lfloor e\rfloor = \{e\}\} \\ & \cup\ \{e \mid e \in \mathcal{E}^{v_k} \wedge \lfloor e\rfloor \subseteq \{e\} \cup \|\theta\|\}\end{array}$$

       The proof is completed by noting that

       $${}^\circ\{e \mid e \in \mathcal{E}^{v_k} \wedge e \notin \mathbb{C} \wedge \lfloor e\rfloor = \{e\}\} = (b_k \cap P \setminus {}^\circ\mathbb{C})$$

       and ${}^\circ\{e \mid e \in \mathcal{E}^{v_k}$ and $\lfloor e\rfloor \subseteq \{e\} \cup \|\theta\|\} = {}^\circ\theta$.
    2. Take t such that ${}^\bullet t = \mathbb{C}_t$. Then, $\mathbb{C}_t \subseteq b \cap P$. By Theorem 1, there cannot be t' enabled at b and ${}^\bullet t' \neq {}^\circ\mathbb{C}_t$ and ${}^\circ\mathbb{C}_t \cap {}^\bullet t' \neq \emptyset$. By using inductive hypothesis (1), we conclude that all events in direct conflict with $\mathbb{C}_t$ in $\mathcal{E}^v$ are in $\mathbb{C}$. Hence, $\|\theta\|$ is a stopped configuration of $\mathcal{E}^v$.
  - $t_{k+1} = \overline{\mathbf{p}} \to (T'', \overline{\mathbb{C}^\circ \setminus (N_\mathbb{C} \ominus p)^\circ})$ for some $\mathbb{C}$, $p \in {}^\circ\mathbb{C}$, and $(T'', \emptyset) = [\![N_\mathbb{C} \ominus p]\!]$. Then $T = T' \cup T''$.
    1. Immediate because $b_k \cap P = b \cap P$.
    2. It follows analogously to the previous case.

$\square$

Lemma 8. *Let $N = (P, T, F, m)$ and $\mathcal{E}$ the event structure of N. If v is recursively stopped configuration and $v = \bigcup_{1\leq i\leq n} v_i$ is a valid decomposition, then*

1. *$[\![N]\!] \xRightarrow{t_1\cdots t_n} (T, b)$ and $v = \bigcup_{1\leq i\leq n}\|t_i\|$;*
2. *$b \cap P = {}^\circ\{e \mid e \in \mathcal{E}^v$ and $\lfloor e\rfloor = \{e\}\}$;*
3. *If v' is a stopped configuration of $\mathcal{E}^v$, then there exists $t \in T$ s.t. $(T, b) \xrightarrow{t}$ and $\|t\| = v'$;*
4. *For all $e \in \mathcal{E}$, if $e \notin (\mathcal{E}^v \cup v)$ implies $\overline{{}^\bullet e} \cap b = 0$.*

Proof. If follows by induction on the length n of the decomposition $v = \bigcup_{1\leq i\leq n} v_i$.

- **Base case (n=0)**. Then, $v = \emptyset$ and $\mathcal{E}^v = \mathcal{E}$. Moreover, $b = m$. Then
  1. It is immediate because $(T, b) = [\![N]\!]$ and $m = b$.

2. Since $b = m$, b corresponds to the preset of all minimal events of $\mathcal{E}^{\emptyset} = \mathcal{E}$.
3. If v' is a stopped configuration of $\mathcal{E}$, then there exists $\mathbb{C} \in \textsc{bc}(N)$ such that $v' \subseteq \mathbb{C}$. Since v' is a maximal configuration, there exists $\theta : \mathbb{C}$ such that $ev(\theta) = v'$. Hence, there exists $t \in T$ such that $\|t\| = v'$. Since, v' is part of an initial prefix, ${}^\bullet t = {}^\circ\mathbb{C} \subseteq m$. Hence, t is enabled.
4. It trivially holds because there does not exist $e \in \mathcal{E}$ and $e \notin (\mathcal{E}^v \cup v)$.

- **Inductive case (n = k+1)**. Take $v' = \bigcup_{1 \leq i \leq k+1} v_i$ and $v = v_{k+1} \cup v'$. Then, (1) $[\![N]\!] \xRightarrow{t_1 \cdots t_k} (T_k, b_k)$ and $v' = \bigcup_{1 \leq i \leq k} \|t_i\|$; and (2) $b_k \cap P = {}^\circ\{e \mid e \in \mathcal{E}^{v'} \text{ and } \lfloor e \rfloor = \{e\}\}$; and (3) If v'' is a stopped configuration of $\mathcal{E}^{v_k}$, then there exists $t \in T$ s.t. $(T_k, b_k) \xrightarrow{t}$ and $\|t\| = v''$; and (4) For all $e \in \mathcal{E}$, if $e \notin (\mathcal{E}^{v'} \cup v')$ implies $\overline{{}^\bullet e} \cap b = 0$.

  By inductive hypothesis (3), there exists t_{k+1} such that $(T_k, b_k) \xrightarrow{t_{k+1}}$ and $\|t_{k+1}\| = v_{t_{k+1}}$. Then, take $(T_k, b_k) \xrightarrow{t} (T', b')$. By using Lemma 7, we conclude that there exists (T_{k+1}, b_{k+1}) such that $(T_k, b_k) \xRightarrow{t_{k+1}} (T_{k+1}, b_{k+1})$ where:

  (a) $b_{k+1} \cap P = b_k \cap P$;

  (b) for all p, q, if $\overline{\mathbf{p}} \in b_k$ and $p \preceq q$, then $\overline{\mathbf{q}} \in b_{k+1}$;

  (c) for all $\mathbb{C} \in \textsc{bc}(N)$ and $\overline{\mathbf{Q}} \subseteq \overline{\mathbf{P}}$, if $\overline{\mathbf{Q}} \subseteq b_{k+1}$ then for all $\mathbb{C}' \in \textsc{bc}(\underline{N_{\mathbb{C}} \ominus Q})$ and $\theta : \mathbb{C}'$ there exists $t \in T_{k+1}$ such that $t = {}^\circ\mathbb{C}' \longrightarrow (\emptyset, \theta^\circ \cup \overline{\mathbb{C}'^\circ \setminus \theta^\circ})$.

  Then,

  1. It follows immediately because $\|t_{k+1}\| = v_{t_{k+1}}$;
  2. Then $t_{k+1} = {}^\circ\mathbb{C} \to (\emptyset, \theta^\circ \cup \overline{\mathbb{C}^\circ \setminus \theta^\circ})$. $b' \cap P = (b_k \cap P \setminus {}^\circ\mathbb{C}) \cup \theta^\circ$.

     $$\begin{aligned} \{e \mid e \in \mathcal{E}^v \wedge \lfloor e \rfloor = \{e\}\} = \quad & \{e \mid e \in \mathcal{E}^{v'} \wedge e \notin \mathbb{C} \wedge \lfloor e \rfloor = \{e\}\} \\ & \cup \{e \mid e \in \mathcal{E}^{v'} \wedge \lfloor e \rfloor \subseteq \{e\} \cup \|\theta\|\} \end{aligned}$$

     The proof is completed by noting that ${}^\circ\{e \mid e \in \mathcal{E}^{v_k} \text{ and } e \notin \mathbb{C} \text{ and } \lfloor e \rfloor = \{e\}\} = (b_k \cap P \setminus {}^\circ\mathbb{C})$ and ${}^\circ\{e \mid e \in \mathcal{E}^{v_k} \text{ and } \lfloor e \rfloor \subseteq \{e\} \cup \|\theta\|\} = {}^\circ\theta$ and by using (a) above.
  3. It follows from (c).
  4. It follows from (b).

$\square$