Pre-nets, read arcs and unfolding: a functorial presentation*

Paolo Baldan[1], Roberto Bruni[2], and Ugo Montanari[2]

[1] Dipartimento di Informatica, Università Ca' Foscari di Venezia, Italia.
[2] Dipartimento di Informatica, Università di Pisa, Italia.
baldan@dsi.unive.it, bruni@di.unipi.it, ugo@di.unipi.it

Abstract. Pre-nets have been recently proposed as a means of providing a functorial algebraic semantics to Petri nets (possibly with read arcs), overcoming some previously unsolved subtleties of the classical model. Here we develop a functorial semantics for pre-nets following a sibling classical approach based on an unfolding construction. Any pre-net is mapped to an acyclic branching net, representing its behaviour, then to a prime event structure and finally to a finitary prime algebraic domain. Then the algebraic and unfolding view are reconciled: we exploit the algebraic semantics to define a functor from the category of pre-nets to the category of domains that is shown to be naturally isomorphic to the unfolding-based functor. All the results are extended to pre-nets with read arcs.

Introduction

P/T Petri nets [Rei85] are one of the most widely known models of concurrency. Since their introduction, almost fifty years ago [Pet62], the conceptual simplicity of the model and its intuitive graphical presentation have attracted the interest of both theoreticians and practitioners. Nevertheless, the concurrent semantics of Petri nets still presents several aspects that cannot be considered fully encompassed. The aim of this paper is to point out the missing fragments of the overall picture and to fill as many gaps as possible, providing neat mathematical constructions.

We concentrate on the semantic interpretation arising from the so-called *individual token philosophy* (ITPh) as opposed to the *collective token philosophy* (CTPh). The two terminologies have been introduced in [GP95] to distinguish the interpretation of tokens in the same place as anonymous and indistinguishable resources (CTPh), from the view of tokens as resources uniquely characterised by their histories and causal dependencies (ITPh). On the one hand, the CTPh, taking the paradigm of multiset rewriting to the extreme consequence is somehow simpler: repeated elements in a multiset are completely equivalent and cannot be distinguished one from the other. On the other hand, the CTPh is less amenable to the full variety of concurrent semantic frameworks that can be studied in the ITPh. Roughly these can be classified in process-oriented, unfolding, algebraic, and logical:

* Research supported by the FET-GC Project IST-2001-32747 AGILE and by the MIUR Project COFIN 2001013518 COMETA. The second author is also supported by an Italian CNR fellowship for research on Information Sciences and Technologies, and by the CS Department of the University of Illinois at Urbana-Champaign.

- The *process* approach focuses on non-sequential / concurrent models of computations and on their composition. Several notions of (deterministic) process have been proposed that rely on different abstractions in modelling resources, executed events and concurrent computations [BD87,GR83,DMM96,Sas98].
- The *unfolding approach* is built on top of nondeterministic processes to account for a broader view of computations, which includes concurrency, causality and conflict. Starting from the seminal work of Winskel [Win87], which focuses on the simpler class of *safe* nets, several authors have contributed to the generalisation of the approach to the full class of P/T Petri nets [MMS92,MMS97a,MMS96], showing that a chain of adjunctions (coreflections in the case of safe or semi-weighted nets) leads from **PTNets** to **PES**, for **PTNets** the category of P/T Petri nets and **PES** the category of prime event structures, which is equivalent to the category **Dom** of coherent finitary prime algebraic domains (for this reason, the unfolding approach is sometimes referred to as a *denotational semantics*).
- The *algebraic approach*, originally proposed in [MM90] for the CTPh under the statement "Petri nets are monoid", recasts the process approach in universal algebras: The idea is to characterise the concurrent model of computation as the initial model in a suitable algebra of decorated computations.
- The *logical view* tries to recast the algebraic approach into deduction theories, whose sentences denote concurrent execution strategies and whose theorems select admissible computations [BMMS01].

Category theory has been shown instrumental in all the above approaches: processes come naturally equipped with notion of a parallel and a sequential composition, which provides the structure of a *monoidal category*; adjunctions and coreflections are categorical notion used in the unfolding semantics to guarantee that all constructions are as good as possible; P/T Petri nets are essentially graphs with structured nodes, and, as such, can be naturally equipped with structure-preserving homomorphism, which can also be seen as simulation morphisms; initiality in the algebraic semantics is again a categorical notion for selecting the best candidate model; finally, the logical view exploits the fact that adjunctions between the categories of models of two theories, like the theory of Petri nets and the theory of concurrent models, can be more conveniently expressed as theory morphisms (whose existence is easier to prove).

When categories are involved, a central property of the semantic constructions, witnessing their appropriateness, is *functoriality*, i.e., the fact that simulation morphisms between nets are preserved at the level of computational, algebraic, logical and denotational models. A second crucial property is *universality*, in the sense of constructions expressed as adjunctions. In fact, we remind that when functors are left/right adjoints they preserve colimits/limits yielding good compositionality properties.

For the ITPh the unfolding approach is completely stable and satisfactory. Instead, the application of the algebraic approach to the ITPh presents several problems basically related to the fact that the monoidal operation on computations is commutative only up to a symmetry natural isomorphism. As a consequence, the construction proposed in [DMM96] fails to preserves some ordinary simulation morphisms between nets. The situation is improved in [MMS97b] up to a pseudo-functorial construction [Sas98]. Correspondingly, different notions of deterministic processes, which differ just in the decoration of minimal and maximal places have been proposed as “concrete” models. The lack of functoriality has also discouraged the formulation of a logical semantics.

The problem intuitively resides in the dichotomy between the multiset view of a state and the need of distinguishing uniquely its elements to track their causal history. A relevant advance of the theory has been the introduction of *pre-nets* [BMMS99] (see also [BMMS01] for an extensive discussion), a variant of ordinary nets where a total ordering is imposed on the places occurring in the pre- and post-set of transitions. Any pre-net can be seen as a concrete “implementation” of the Petri net obtained by forgetting about the ordering of places in pre- and post-sets. Using strings rather than multisets allows to uniquely characterise each element by its position. Thus pre-nets allow to obtain a satisfactory algebraic treatment, where the construction of the model of computation yields an adjunction between the category of pre-nets and the category of models (symmetric monoidal categories [Mac71]) and it can be expressed as a theory morphism, accounting for the algebraic and logical views. Notably, the construction of the model of computation for all pre-nets implementing the same Petri net yields the same result, hence we can define the semantics of a net as *the* algebraic semantics of any of its pre-net implementations. Still the picture is incomplete, since some classical approaches to the semantics of Petri nets have not been yet explored for pre-nets.

In this paper we complete the theory of pre-nets by showing that:

- Concrete notions of deterministic occurrence pre-nets and of pre-net processes can be defined in analogy with Petri nets. Finite processes form a symmetric monoidal category which turns out to be isomorphic via a symmetric monoidal functor to the algebraic model of computation, thus reconciling the process and algebraic view in a fully functorial construction (a result not possible for Petri nets). Moreover, a graphical presentation is introduced for pre-net processes.
- A domain semantics for pre-nets can be defined by generalising a construction proposed for ordinary nets in [MMS96]. Given a pre-net R, the comma category $\langle u \downarrow \mathcal{Z}(R) \rangle$, where u is the initial state of R and $\mathcal{Z}(R)$ its algebraic model, is a preorder whose ideal completion is a prime algebraic domain. Roughly this domain consists of the set of deterministic processes of the net, endowed with a kind of prefix ordering.
- An unfolding semantics can be defined which associates to any pre-net, first an acyclic pre-net representing all its possible computations in a single branching structure, then an event structure and finally a prime algebraic domain.
- Since the unfolding is essentially a nondeterministic process that completely describes the behaviour of a pre-net, a clear link between the unfolding and the algebraic approach is called for. The result showing that the domain originating from the algebraic model of computation and the one extracted from the unfolding are isomorphic, can now be stated in a satisfactory categorical framework: the two constructions can be expressed as *naturally isomorphic functors* (while the analogous result for ordinary Petri nets [MMS96] holds only at the level of objects).
- Finally, the pre-net and Petri net framework are reconciled by explaining how the domain semantics of a net and of its pre-net implementations are related.

We remark that, although in the case of pre-nets all the construction are functorial, one link is still missing, because the functor that abstracts the unfolding of a pre-net to a prime event structure is not characterised as a universal construction. Whether the mentioned construction can be defined as a right adjoint or not is a non-trivial question. We strongly conjecture that the answer is negative, but this is left as an open problem.

Along the years, Petri nets have been generalised in several ways to increase their expressivity. In the last part of the paper we focus on a mild but significant extension, i.e., the addition of *read arcs*, which allows to provide a faithful representation of read-only accesses to resources. Nets with read arcs, called *contextual nets* in [MR95], have been used to model a variety of applications and phenomena, such as transaction serializability in databases [DFMR94], concurrent constraint programming [MR94], asynchronous systems [Vog97], and analysis of cryptographic protocols [CW01].

Pre-nets have been already shown to be useful to define a neat algebraic semantics for contextual nets [BMMS02]. Here, relying on some previous work on the different semantic approaches for nets with read arcs, we discuss how the whole theory developed in this paper for ordinary pre-nets generalises in the presence of read arcs.

Synopsis. The rest of the paper is structured as follows. Section 1 reviews the basics of pre-nets and their algebraic semantics. Section 2 defines a process semantics for pre-nets and compares it to the algebraic semantics. Section 3 develops the unfolding semantics of pre-nets. Section 5 extends our results to nets with read arcs. Finally, Section 6 summarises the results in the paper and some open questions. We assume that the reader has some familiarity with P/T Petri net theory and category theory.

1 Pre-nets and their algebraic semantics

In this section we recall the basics of pre-nets [BMMS99,BMMS01], discussing their algebraic semantics and the relation with ordinary P/T Petri nets.

Notation. Given a set X, we denote by $X^{\otimes}$ the free monoid over X (finite strings of elements of X) with the empty string ε as the unit, and by $X^{\oplus}$ the free commutative monoid over X (finite multisets over X) with unit the empty set $\emptyset$. We write $\mu : X^{\otimes} \to X^{\oplus}$ for the function mapping any string to the underlying multiset. Furthermore, given a function $f : X \to Y^{\otimes}$ we denote by $f^{\otimes} : X^{\otimes} \to Y^{\otimes}$ its obvious monoidal extension. Similarly, given $g : X \to Y^{\oplus}$ we denote by $g^{\oplus} : X^{\oplus} \to Y^{\oplus}$ its commutative monoidal extension. Given $u \in X^{\otimes}$ or $u \in X^{\oplus}$ we denote by $[\![u]\!]$ the underlying subset of X defined in the obvious way. When set relations are used over string and multisets, we implicitly refer to the underlying set. E.g., for $u, v \in X^{\otimes}$ (or $X^{\oplus}$) by $x \in u$ we mean $x \in [\![u]\!]$ and similarly $u \cap v$ means $[\![u]\!] \cap [\![v]\!]$.

Recall that a *P/T Petri net* is a tuple $N = (\partial_0, \partial_1, S, T)$, where S is a set of *places*, T is a set of *transitions*, and $\partial_0, \partial_1 : T \to S^{\oplus}$ are functions assigning multisets called source and target, respectively, to each transition. A *marked* net is a pair $\langle N, m\rangle$ where N is a P/T Petri net and $m \in S^{\oplus}$. A *Petri net morphism* $f = \langle f_{\mathrm{s}}, f_{\mathrm{t}}\rangle : N \to N'$ is a pair where $f_{\mathrm{s}} : S^{\oplus} \to S'^{\oplus}$ is a monoid homomorphism, and $f_{\mathrm{t}} : T \to T'$ is a function such that $\partial'_i \circ f_{\mathrm{t}} = f_{\mathrm{s}} \circ \partial_i$, for any $t \in T$ and $i \in \{0, 1\}$. The category of P/T Petri nets (as objects) and Petri net morphisms (as arrows) is denoted by **Petri**. A morphism of marked P/T nets $f : \langle N, m\rangle \to \langle N', m'\rangle$ is subject to the additional requirement of preservation of the

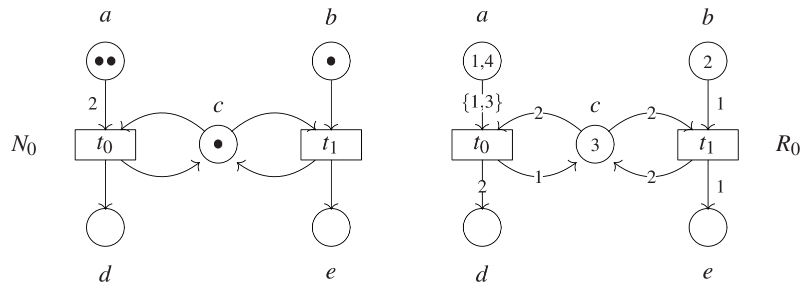

Fig. 1. The P/T Petri net N_0 and (one of) its pre-net implementation R_0.

initial marking, i.e., $f_s(m) = m'$. The category of marked P/T Petri nets (as objects) and marked Petri net morphisms (as arrows) is denoted by $\mathbf{Petri}_*$.

A *pre-net* is roughly a Petri net where the resources (tokens in places) are linearly ordered. In other words, the state as well as the pre- and post-conditions of transitions are strings rather than multisets of places.

Definition 1 (pre-net). *A* pre-net *is a tuple $R = (\zeta_0, \zeta_1, S, T)$, where S is a set of* places*, T is a set of* transitions*, and $\zeta_0, \zeta_1 : T \to S^{\otimes}$ are functions assigning, respectively, source and target to transitions. A* marked *pre-net is a pair $\langle R, u\rangle$ with R a pre-net and $u \in S^{\otimes}$.*

The pictorial representation of Petri nets has certainly played an important role in their large diffusion as a specification framework. This graphical presentation (places represented as circles, transition as boxes, pre- and post-set multirelation as weighted arcs, tokens as black bullets) can be extended to pre-nets by adopting the following conventions: (1) weighted arcs are replaced by arcs labelled with the *ordered list of positions* in which the place appears in the pre- / post-set of the transition, with lists of length greater than one enclosed in curly brackets; (2) tokens are represented as numbers denoting their positions in the current state. An example of pre-net R_0 can be found in the right part of Fig. 1. It will be used throughout the paper to illustrate definitions and concepts. From the inscription $\{1,3\}$ of the arc from a to t_0, we see that a firing of t_0 requires two tokens from a, to be taken as first and third consumed resources, while the second token to be consumed by t_0 must be taken from c, as imposed by the inscription 2 of the arc from c to t_0 (we remark that 2 denotes a position, not the number of tokens to be consumed). Moreover, from the inscriptions inside the circles for a, b and c, we note that the initial marking of R_0 is the string $u = abca$, i.e., that the a occurs in the first and fourth positions of u, b in the second, and c in the third.

As for P/T Petri nets, the notion of pre-net morphism naturally arises from an algebraic view, where places and transitions play the role of sorts and operators.

Definition 2 (pre-net morphism). *A pre-net morphism from $R = (\zeta_0, \zeta_1, S, T)$ to $R' = (\zeta'_0, \zeta'_1, S', T')$ is a pair $f = \langle f_s, f_t\rangle$ where $f_s : S^{\otimes} \to S'^{\otimes}$ is a monoid homomorphism, and $f_t : T \to T'$ is a function such that $\zeta'_i \circ f_t = f_s \circ \zeta_i$, for $i \in \{0,1\}$. We denote by* **PreNet** *the category of pre-nets and their morphisms with the obvious composition.*

A marked pre-net morphism from $\langle R,u\rangle$ to $\langle R',u'\rangle$ is a pre-net morphism $f : R \to R'$ such that $f_s(u) = u'$. We denote by **PreNet**$_*$ *the category of marked pre-nets and their morphisms with the obvious composition.*

Pre-nets can be seen as a specification formalism (slightly) more concrete than Petri nets. In particular any pre-net R can be thought of as an "implementation" of the Petri net which is obtained from R replacing any string by the corresponding multiset. This construction is formalised below.

Definition 3. *The functor $\mathcal{A}$:* **PreNet** $\to$ **Petri** *is defined as follows:*

- *any pre-net $R = (\zeta_0,\zeta_1,S,T)$ is mapped to $\mathcal{A}(R) = (\partial_0,\partial_1,S,T)$, where $\partial_i(t) = \mu(\zeta_i(t))$ for each $t \in T$ and $i \in \{0,1\}$;*
- *any pre-net morphism $f : R \to R'$ is mapped to $\mathcal{A}(f) = \langle g_{\mathrm{s}}^{\oplus}, f_{\mathrm{t}}\rangle$, where $g_{\mathrm{s}}(s) = \mu(f_{\mathrm{s}}(s))$ for each $s \in S$.*

We denote by $\mathcal{A}_*$: **PreNet**$_*$ $\to$ **Petri**$_*$ *the obvious extension of $\mathcal{A}$ to marked nets.*

For instance, referring to Fig. 1, the ordinary Petri net N_0 in the left part is implemented by R_0, i.e., we have $\mathcal{A}_*(R_0) = N_0$. The transition $t_0 : 2a \oplus c \to c \oplus d \in N_0$ is implemented as $t_0 : aca \to cd \in R_0$, and $t_1 : b \oplus c \to c \oplus e \in N_0$ as $t_1 : bc \to ec \in R_0$. Clearly alternative implementations would have been possible exploiting different linearizations.

Intuitively, a computation of a pre-net consists of "explicit" steps, namely firings of transitions which consume and produce resources, and of "implicit" steps which rearrange the order of the resources to allow the application of transitions. All the sequences of implicit steps that implement the same permutation of a given state are indistinguishable. Formally, the model of computation of a pre-net is the free symmetric strict monoidal category generated by the pre-net, the symmetries playing the role of the above mentioned implicit steps. Let **SSMC** be the category of symmetric strict monoidal categories (as objects) and symmetric monoidal functors (as arrows), and let **SSMC**$^{\otimes}$ denote the full subcategory containing only the categories whose monoid of objects is freely generated. Then the algebraic model of computation of a pre-net R is its image $\mathcal{Z}(R)$ through $\mathcal{Z}$: **PreNet** $\to$ **SSMC**$^{\otimes}$, the left adjoint to the obvious forgetful functor from **SSMC**$^{\otimes}$ to **PreNet**. A more illustrative definition is given below.

Definition 4. *Given a pre-net $R = (\zeta_0,\zeta_1,S,T)$, the model of computation $\mathcal{Z}(R)$ is a symmetric monoidal category whose objects are the elements of $S^{\otimes}$ and whose arrows are generated by the rules in Fig. 2, quotiented out by the axioms of monoidal categories and the coherence axioms making $\gamma_{\_,\_}$ the symmetry natural isomorphism (all axioms are collected in Fig. 3).*

Recall that a *pointed* category is a pair $\langle \mathbf{C}, O_{\mathbf{C}}\rangle$, where $\mathbf{C}$ is a category and $O_{\mathbf{C}}$ is an object in $\mathbf{C}$. A *pointed functor* $F : \langle \mathbf{C}, O_{\mathbf{C}}\rangle \to \langle \mathbf{D}, O_{\mathbf{D}}\rangle$ is a functor $F : \mathbf{C} \to \mathbf{D}$ such that $F(O_{\mathbf{C}}) = O_{\mathbf{D}}$. The construction of the model of computation extends to marked pre-nets and to the category **SSMC**$^{\otimes}_*$ of pointed strictly symmetric monoidal categories.

Definition 5. *Given a marked pre-net $\langle R,u\rangle$, the* model of computation *$\mathcal{Z}_*(\langle R,u\rangle)$ is the pointed category $\langle \mathcal{Z}(R),u\rangle$.*

$$\frac{u \in S^{\otimes}}{id_u : u \to u \in \mathcal{Z}(R)} \qquad \frac{u, v \in S^{\otimes}}{\gamma_{u,v} : uv \to vu \in \mathcal{Z}(R)} \qquad \frac{t \in T \quad \zeta_0(t) = u \quad \zeta_1(t) = v}{t : u \to v \in \mathcal{Z}(R)}$$

$$\frac{\alpha : u \to v, \quad \alpha' : u' \to v' \in \mathcal{Z}(R)}{\alpha \otimes \alpha' : uu' \to vv' \in \mathcal{Z}(R)} \qquad \frac{\alpha : u \to v, \quad \beta : v \to w \in \mathcal{Z}(R)}{\alpha;\beta : u \to w \in \mathcal{Z}(R)}$$

Fig. 2. Inference rules for $\mathcal{Z}(R)$.

For any $u, v, w \in S^{\otimes}$ and
for any $\alpha : u \to v, \beta : v \to w, \delta : w \to z, \alpha' : u' \to v', \beta' : v' \to w', \alpha'' : u'' \to v'' \in \mathcal{Z}(R)$:

UNIT:	$id_\varepsilon \otimes \alpha = \alpha = \alpha \otimes id_\varepsilon,$	
ASSOCIATIVITY:	$(\alpha \otimes \alpha') \otimes \alpha'' = \alpha \otimes (\alpha' \otimes \alpha'')$	$(\alpha;\beta);\delta = \alpha;(\beta;\delta)$
IDENTITIES:	$\alpha; id_v = \alpha = id_u; \alpha$	$id_u \otimes id_v = id_{uv}$
FUNCTORIALITY:	$(\alpha;\beta) \otimes (\alpha';\beta') = (\alpha \otimes \alpha');(\beta \otimes \beta')$	
NATURALITY:	$(\alpha \otimes \alpha');\gamma_{v,v'} = \gamma_{u,u'};(\alpha' \otimes \alpha)$	
COHERENCE:	$\gamma_{u,vw} = (\gamma_{u,v} \otimes id_w);(id_v \otimes \gamma_{u,w})$	$\gamma_{u,v};\gamma_{v,u} = id_{uv}$

Fig. 3. Axioms for $\mathcal{Z}(R)$.

Notice that $\mathcal{Z}_*$ extends to a left adjoint functor from $\mathbf{PreNet}_*$ to $\mathbf{SSMC}^{\otimes}_*$.

Given a pre-net R and two states $u, v \in S^{\otimes}$ we say that v is reachable from u if there is an arrow $\alpha : u \to v$ in $\mathcal{Z}(R)$. If $\langle R, u \rangle$ is a marked pre-net we say that v is *reachable* if it is reachable from u. One can easily see that v is reachable in $\langle R, u \rangle$ if and only if $\mu(v)$ is reachable in $\mathcal{A}_*(\langle R, u \rangle)$. More generally, given any P/T net N, all its pre-net implementations have essentially the same behaviour, in the sense that they have isomorphic models of computation. Hence the semantics of N can be recovered by an arbitrarily chosen pre-net implementation.

Theorem 1. *For any pair of pre-nets R and R', if $\mathcal{A}(R) \simeq \mathcal{A}(R')$ then $\mathcal{Z}(R) \simeq \mathcal{Z}(R')$ via a symmetric monoidal functor.*

Moreover, the category $\mathcal{Z}(R)$ can be quotiented out by suitable axioms to recover all the algebraic computational models of $\mathcal{A}(R)$ in the literature (e.g. concatenable processes, commutative processes). Analogous results holds also in the marked case.

2 Concatenable processes for pre-nets

In this section we introduce a notion of (concatenable) process for pre-nets. A process is intended to provide a static representation of a concurrent computation, which makes explicit the events occurring in the computation and their causal dependencies. The appropriateness of our notion of pre-net process will be formalised by showing that for any pre-net the category of concatenable processes is isomorphic to its model of computation via a symmetric monoidal functor.

2.1 Safe and occurrence pre-nets

Let R be a pre-net. A state $u \in S^{\otimes}$ is called *safe* if any place occurs at most once in u, i.e., if $\mu(u)$ is a safe marking. A marked pre-net is called safe if the source and target of all transitions as well as all the reachable states are safe.

Definition 6 (causality, conflict, concurrency). *Let $R = (\zeta_0, \zeta_1, S, T)$ be a pre-net. The* causality relation *is the least transitive relation $<_R \subseteq (S \cup T) \times (S \cup T)$ such that*

$$\text{(i) if } s \in \zeta_0(t) \text{ then } s <_R t; \quad \text{(ii) if } s \in \zeta_1(t) \text{ then } t <_R s.$$

Given a place or transition $x \in S \cup T$, we denote by $\lfloor x \rfloor$ the set of causes *of x in T, defined as $\lfloor x \rfloor = \{t \in T \mid t \leq_R x\} \subseteq T$, where $\leq_R$ is the reflexive closure of $<_R$.*

The conflict relation *$\#_R \subseteq (S \cup T) \times (S \cup T)$ is defined as the least relation such that*

$$\text{(i) if } \zeta_0(t) \cap \zeta_0(t') \neq \emptyset \text{ then } t \#_R t'; \quad \text{(ii) if } x \#_R x' \text{ and } x' \leq_R x'' \text{ then } x \#_R x''.$$

A set of places $X \subseteq S$ is concurrent, *written $co(X)$ if for any $s, s' \in X$ neither $s < s'$ nor $s \# s'$, and $\bigcup_{x \in X} \lfloor x \rfloor$ is finite.*

Definition 7 (occurrence pre-net). *An* occurrence pre-net *is a safe pre-net R such that (i) causality $<_R$ is a partial order and, for any transition t, the set of causes $\lfloor t \rfloor$ is finite; (ii) there are no backward conflicts, i.e., for any $t \neq t'$, $\zeta_1(t) \cap \zeta_1(t') = \emptyset$; (iii) conflict $\#_R$ is irreflexive. An occurrence pre-net is* deterministic *if it has no forward conflicts, i.e., for any $t \neq t'$, $\zeta_0(t) \cap \zeta_0(t') = \emptyset$.*

We denote by $\mathbf{PreOcc}_*$ *the full subcategory of* $\mathbf{PreNet}_*$ *whose objects are occurrence pre-nets.*

It is immediate to verify that the relations of causality and conflict in a pre-net R are the same as in the implemented Petri net $\mathcal{A}(R)$. Hence R is a safe (occurrence) pre-net if and only if the corresponding Petri net $\mathcal{A}(R)$ is a safe (occurrence) net. This implies that $\mathcal{A}_*$ restricts to a well-defined functor from $\mathbf{PreOcc}_*$ to $\mathbf{Occ}_*$, the full subcategory of $\mathbf{Petri}_*$ where objects are occurrence nets.

2.2 Processes of a pre-net

An interesting feature of Petri nets is the fact that a net process can still be represented as a special Petri net (decorated with a morphism to the original net) [GR83]. This is true also for pre-nets.

Let us call a pre-net morphism $f : R \to R'$ *elementary* if for any $s \in S$, $f_s(s) \in S'$ (places are sent to single places rather than to strings).

Definition 8 (process). *Let $R = (\zeta_0, \zeta_1, S, T)$ be a pre-net. A* process *of R is an elementary pre-net morphism $\pi : O \to R$ where O is an occurrence pre-net and for any $t, t' \in T_O$, if $f_t(t) = f_t(t')$ and $\zeta_0(t) = \zeta_0(t')$ then $t = t'$ (irredundancy).*

The process π is finite / deterministic if the underlying occurrence pre-net O is finite / deterministic. For a finite deterministic process π we denote by $\min(\pi)$ *(resp.,* $\max(\pi)$*) the set of places of O which are minimal (resp., maximal) w.r.t. $\leq_O$.*

A concatenable process of a pre-net is a deterministic finite process of the net with explicit source and target states, i.e., with a total ordering in the minimal and maximal places of the underlying occurrence pre-net.

Definition 9 (concatenable process). *A concatenable process of a pre-net R is a triple $\delta = \langle\sigma,\pi,\tau\rangle$, where π is a deterministic finite process of R and $\sigma,\tau \in S_O^{\otimes}$ are string of places in S_O such that*

$$\mu(\sigma) = \min(\pi) \qquad \text{and} \qquad \mu(\tau) = \max(\pi).$$

We denote by $\zeta_0(\delta)$ the string $\pi_s^{\otimes}(\sigma)$ and by $\zeta_1(\delta)$ the string $\pi_s^{\otimes}(\tau)$.

An isomorphism of (concatenable) processes δ and δ' is an isomorphism of the underlying pre-nets consistent with the mapping to the original pre-net and with the linearizations of minimal and maximal places. The isomorphism class of a concatenable process δ is written $[\delta]$ and called an *abstract* concatenable process.

Concatenable processes $\delta = \langle\sigma,\pi,\tau\rangle$ of pre-nets can be graphically represented by slightly adjusting the visual modelling of ordinary Petri processes: (1) places (and transitions) are labelled by their images through π, (2) minimal and (resp. maximal) places carry also as superscript (resp., subscript) their position in σ (resp., τ); (3) arcs are labelled by the (unique) position in which the place appears in the pre- and post-set of the transition (again, we remark that arc labels stand for positions, not for weights).

In Fig. 4 some simple processes are illustrated (for our running example R_0) that correspond to single transitions, place identities and permutations.

Given two concatenable processes $\delta_1 = \langle\sigma_1,\pi_1,\tau_1\rangle$ and $\delta_2 = \langle\sigma_2,\pi_2,\tau_2\rangle$, such that $\zeta_1(\delta_1) = \zeta_0(\delta_2)$ their concatenation is defined as the process obtained by gluing the maximal places of π_1 and the minimal places of π_2 according to their orderings.

Definition 10 (sequential composition). *Let $\delta_1 = \langle\sigma_1,\pi_1,\tau_1\rangle$ and $\delta_2 = \langle\sigma_2,\pi_2,\tau_2\rangle$ be concatenable processes of a pre-net R such that $\zeta_1(\delta_1) = \zeta_0(\delta_2)$. Suppose $T_1 \cap T_2 = \emptyset$ and $S_1 \cap S_2 = \max(\pi_1) = \min(\pi_2)$, with $\tau_1 = \sigma_2$. In other words δ_1 and δ_2 overlap only on $\max(\pi_1) = \min(\pi_2)$, and such places carry the same ordering in the interfaces τ_1 and σ_2. Then their sequential composition $\delta_1;\delta_2$ is the concatenable process $\delta = \langle\sigma_1,\pi,\tau_2\rangle$, where the process π is the (componentwise) union of π_1 and π_2.*

The above construction induces a well-defined operation of sequential composition between abstract concatenable processes. In particular, if $[\delta_1]$ and $[\delta_2]$ are abstract concatenable processes such that $\zeta_1(\delta_1) = \zeta_0(\delta_2)$ then we can always find $\delta_2' \in [\delta_2]$ such that $\delta_1;\delta_2'$ is defined. Moreover the result of the composition seen at abstract level, namely $[\delta_1;\delta_2']$, does not depend on the particular choice of the representatives.

Definition 11. *We denote by $\mathcal{PP}(R)$ the category having the elements of $S^{\otimes}$ as objects and abstract concatenable processes of R as arrows, with obvious composition as in Definition 10 and obvious identities.*

The category $\mathcal{PP}(R)$ is a symmetric strict monoidal category. In fact (1) parallel composition $\otimes$ is readily defined for processes $\delta_1 = \langle\sigma_1,\pi_1,\tau_1\rangle$ and $\delta_2 = \langle\sigma_2,\pi_2,\tau_2\rangle$ such that $T_1 \cap T_2 = S_1 \cap S_2 = \emptyset$, as $\delta_1 \otimes \delta_2 = \langle\sigma_1\sigma_2,\pi,\tau_1\tau_2\rangle$, where π is the componentwise union of π_1 and π_2; (2) parallel composition induces a well-defined tensor

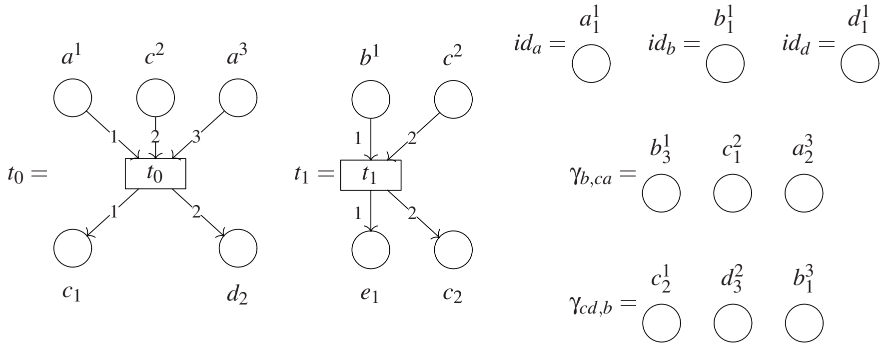

Fig. 4. Textual and graphical representation of simple pre-net processes.

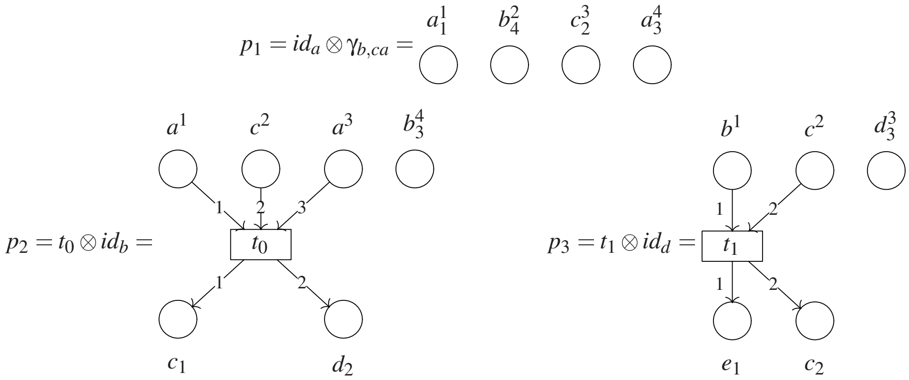

Fig. 5. Tensor product of simple processes.

product between abstract concatenable processes; (3) the tensor product is associative (but not commutative!) and it has the empty process $\langle \varepsilon, \pi_\emptyset, \varepsilon \rangle$ as unit; (4) the component $\gamma_{u,v}$ of the symmetry natural isomorphism is defined by the abstract class of processes $\langle \sigma_u \sigma_v, \pi, \sigma_v \sigma_u \rangle$ with no transitions and such that $\pi^\otimes(\sigma_u) = u$ and $\pi^\otimes(\sigma_v) = v$.

In Fig. 5 the processes of Fig. 4 are composed via tensor products in the larger processes $p_1 : abca \to acab$, $p_2 : acab \to cdb$ and $p_3 : bcd \to ecd$. Finally, in Fig. 6, the processes illustrated so far are composed sequentially in $p_4 : abca \to cdb$, $p_5 : cdb \to ecd$ and $p : abca \to ecd$.

The next theorem shows that pre-net processes provide an appropriate description of the concurrent computations of a pre-net R, in the sense that concatenable pre-net processes can be seen as concrete representatives of the arrows in $\mathcal{Z}(R)$.

Theorem 2. *The category $\mathcal{PP}(R)$ is isomorphic to the model of computation $\mathcal{Z}(R)$ via a symmetric monoidal functor.*

The theorem above is proved by observing that, being $\mathcal{PP}(R)$ a symmetric monoidal category, a functor from $F : \mathcal{Z}(R) \to \mathcal{PP}(R)$ can be easily defined by mapping generators to generators. A functor in the converse direction, is defined by identifying a normal

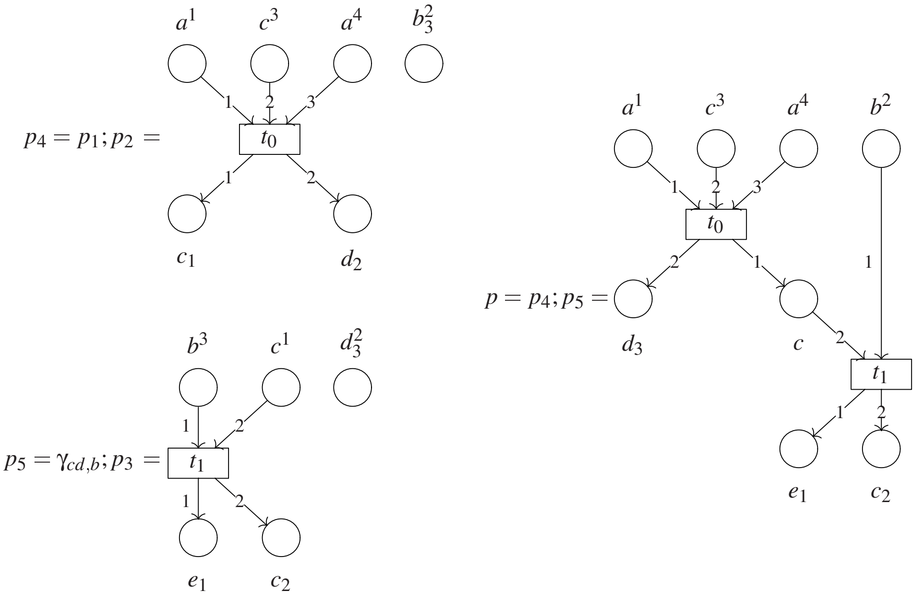

Fig. 6. Sequential composition of processes.

form for the processes in $\mathcal{PP}(R)$ which, roughly, corresponds to a maximally concurrent computation. As a technical remark, the proof is much simpler w.r.t. analogous ones for (concatenable, strongly concatenable) process categories associated to Petri nets, as we can (arbitrarily) fix the normal form expression in such a way that all isomorphic processes have exactly the same normal form (whereas in Petri nets the normal form can be fixed only up-to isomorphism).

3 Unfolding of pre-nets

A deterministic process describes a single deterministic computation of the net. The unfolding approach, originally devised in [NPW81], associates to a system a single denotational structure representing, in an unambiguous way, all the events occurring in any possible computation and their dependencies. This structure expresses not only the causal ordering between the events, but also gives an explicit representation of the branching (choice) points of the computations.

In this section we develop a functorial unfolding semantics for pre-nets, discussing the difficulties which arise in trying to express this functor as a universal construction.

3.1 Unfolding construction

Given a marked pre-net $\langle R, u\rangle$ the unfolding construction unwinds R into an occurrence pre-net, starting from the initial state u, firing transitions in all possible way and recording the corresponding occurrences.

$$\frac{1 \leq i \leq |u|}{u'_i = \langle \emptyset, u_i, i\rangle \in S' \quad \eta_s(u'_i) = u_i} \qquad \frac{v \in S'^{\otimes} \text{ safe} \quad co([v]) \quad t \in T \quad \eta_s^{\otimes}(v) = \zeta_0(t)}{t' = \langle v, t\rangle \in T' \quad \eta_t(t') = t \quad \zeta'_0(t') = v}$$

$$\frac{t' = \langle v, t\rangle \in T' \quad \zeta_1(t) = w_1 \ldots w_n}{w'_i = \langle \{t'\}, w_i, i\rangle \in S' \quad \eta_s(w'_i) = w_i \quad \zeta'_1(t') = w'_1 \ldots w'_n}$$

Fig. 7. Inference rules for the unfolding $\mathcal{U}_p(\langle R, u\rangle)$ of a pre-net R.

Definition 12 (unfolding). *Let $\langle R, u\rangle$ be a marked pre-net. The unfolding $\mathcal{U}_p(\langle R, u\rangle) = ((\zeta'_0, \zeta'_1, S', T'), u')$ and the* folding morphism $\eta_R = \langle \eta_t, \eta_s\rangle : \mathcal{U}_p(R) \to R$ *are the occurrence pre-net and (elementary) pre-net morphism inductively defined by the rules in Fig. 7, with $u' = \langle \emptyset, u_1, 1\rangle \ldots \langle \emptyset, u_{|u|}, |u|\rangle$ (where u_i denotes the ith element of the string u, and $|u|$ is the length of u).*

Observe that items in the unfolding are enriched with their causal histories. Any place $s' = \langle x, w_i, i\rangle$ records its generator x (x is empty when the place is in the initial state, otherwise x is a singleton), the place w_i in the original pre-net and a number i which allow to distinguish multiple occurrences of tokens in the same place, having the same history. Any transition $t' = \langle v, t\rangle$ represents a firing of t that consumes the string of resources v.

The unfolding of our running example R_0, with initial state $abca$, is depicted in Fig. 8. The morphism $\eta_{R_0} : \mathcal{U}_p(\langle R_0, abca\rangle) \to R_0$ is implicitly represented by labelling each place and transition x with its image $\eta_{R_0}(x)$. For some items in the unfolding also the concrete identity is provided. For instance, $a_1 = \langle \emptyset, a, 4\rangle$ represents the occurrence of a in the fourth position of the initial marking, $t'_0 = \langle a_4 c_3 a_1, t_0\rangle$ represent an occurrence of t_0, which fires using the fourth, third and second resource in the initial state.

The unfolding construction can be characterised as a universal construction establishing a coreflection between the categories $\mathbf{PreOcc}_*$ and $\mathbf{PreNet}_*$

Theorem 3. *The unfolding construction induces a functor $\mathcal{U}_p : \mathbf{PreNet}_* \to \mathbf{PreOcc}_*$, right adjoint to the inclusion $I_p : \mathbf{PreOcc}_* \to \mathbf{PreNet}_*$, with counit $\eta : I_p \circ \mathcal{U}_p \to 1$.*

3.2 Event structure and domain semantics

The unfolding semantics for a pre-net can be naturally abstracted to a prime event structure semantics. *Prime event structures* (PES) are a simple event based model of (concurrent) computations in which events are considered as atomic and instantaneous steps, which can appear only once in a computation. An event can occur only after some other events (its *causes*) have taken place and the execution of an event can inhibit the execution of other events. This is formalised via two binary relations: *causality*, modelled by a partial order relation, and *conflict*, modelled by a symmetric and irreflexive relation, hereditary w.r.t. causality.

Definition 13 (prime event structures). *A* prime event structure *(PES) is a tuple $P = \langle E, \leq, \#\rangle$, where E is a set of* events *and $\leq$, $\#$ are binary relations on E called* causality *and* conflict*, respectively, such that:*

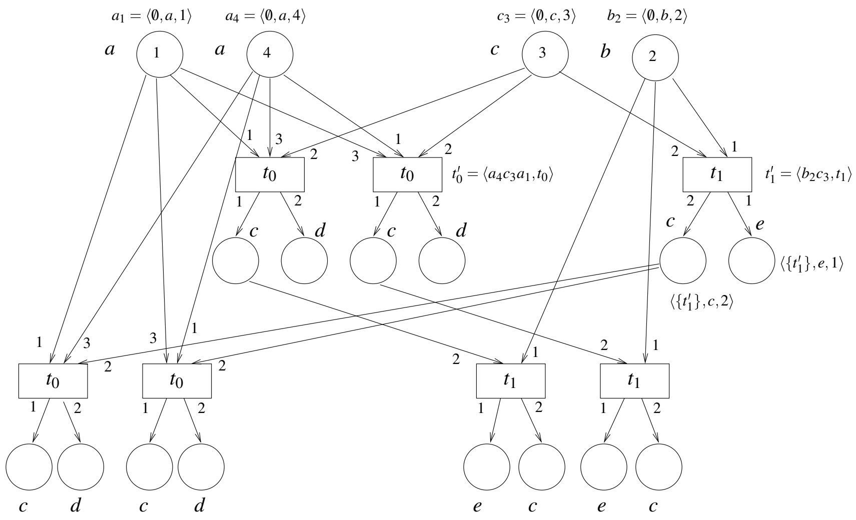

Fig. 8. The unfolding of $\langle R_0, abca\rangle$.

1. *the relation $\leq$ is a partial order and $\lfloor e \rfloor = \{e' \in E : e' \leq e\}$ is finite for all $e \in E$;*
2. *the relation # is irreflexive, symmetric and hereditary with respect to $\leq$, i.e., $e\#e'$ and $e' \leq e''$ implies $e\#e''$ for all $e, e', e'' \in E$;*

Let $P_0 = \langle E_0, \leq_0, \#_0\rangle$ and $P_1 = \langle E_1, \leq_1, \#_1\rangle$ be two PES*'s. A* PES-morphism $f : P_0 \to P_1$ *is a partial function $f : E_0 \to E_1$ such that for all $e_0, e'_0 \in E_0$, assuming that $f(e_0)$ and $f(e'_0)$ are defined:*

1. *$\lfloor f(e_0) \rfloor \subseteq f(\lfloor e_0 \rfloor)$;*
2. *(a) $f(e_0) = f(e'_0) \wedge e_0 \neq e'_0 \Rightarrow e_0\#_0e'_0$;* *(b) $f(e_0)\#_1 f(e'_0) \Rightarrow e_0\#_0e'_0$;*

The category of prime event structures and PES*-morphisms is denoted by* **PES**.

Given an occurrence pre-net the corresponding PES can be obtained by forgetting about the places, keeping the transitions and the dependency relations among them. The transformation is functorial since the transition component of a morphism between occurrence pre-nets satisfies the requirements to be a PES-morphism between the underlying PES's.

Definition 14 (from occurrence pre-nets to PES's). *Let $\mathcal{E}_p$:* **PreOcc**$_*$ $\to$ **PES** *be the functor defined on objects by $\mathcal{E}_p(R) = \langle T, \leq_R, \#_R\rangle$ for any occurrence pre-net R and on arrows by $\mathcal{E}_p(f) = f_t$ for each occurrence pre-net morphism $f : R_0 \to R_1$.*

Winskel in his seminal work [Win87] shows that PES's are intimately connected with another classical semantical model, i.e., *prime algebraic, finitely coherent, finitary*

$$\begin{tikzcd}
\mathbf{PreNet}_* \arrow[r, shift right, "\mathcal{U}_p"'] \arrow[r, phantom, "\perp"] & \mathbf{PreOcc}_* \arrow[l, hook', shift right, "I_p"'] \arrow[r, "\mathcal{E}_p"'] & \mathbf{PES} \arrow[r, shift right, "\mathcal{L}"'] \arrow[r, phantom, "\perp"] & \mathbf{Dom} \arrow[l, shift right, "\mathcal{P}"']
\end{tikzcd}$$

Fig. 9. Denotational semantics of pre-nets.

---

partial orders, hereafter referred to simply as *domains* [Ber78]. Formally, an equivalence is established between the category **PES** of prime event structures and the category **Dom** of domains and additive, stable, immediate precedence-preserving functions:

$$\begin{tikzcd}
\mathbf{PES} \arrow[r, shift right, "\mathcal{L}"'] \arrow[r, phantom, "\sim"] & \mathbf{Dom} \arrow[l, shift right, "\mathcal{P}"']
\end{tikzcd}$$

The functor $\mathcal{L}$ associates to each PES the domain of its configurations, while the functor $\mathcal{P}$ maps each domain to a PES having its prime elements as events. Relying on this classical result, the PES semantics defined in this section for pre-nets can be equivalently interpreted as a domain semantics. The situation is summarised in Fig. 9.

Interestingly, a clear relation can be established between the functorial domain semantics of a Petri net N as defined in [MMS96] and the domain semantics of its pre-net implementations defined here. Recall that, generalising Winskel's work on safe nets [Win87], the semantics for ordinary P/T Petri nets in [MMS96] is given as a chain of adjunctions from the category of nets to the category of domains. The diagram below summarises these results.

$$\begin{tikzcd}
\mathbf{PTNets}_* \arrow[r, shift right, "\mathcal{U}_d"'] \arrow[r, phantom, "\perp"] & \mathbf{DecOcc}_* \arrow[l, shift right] \arrow[d, shift right, "\mathcal{F}"'] \arrow[d, phantom, "\vdash"] & & \\
\mathbf{Safe}_* \arrow[r, shift right, "\mathcal{U}"'] \arrow[r, phantom, "\perp"] & \mathbf{Occ}_* \arrow[l, hook', shift right] \arrow[u, shift right, "\mathcal{D}"'] \arrow[r, shift right, "\mathcal{E}"'] \arrow[r, phantom, "\perp"] & \mathbf{PES} \arrow[l, shift right, "\mathcal{N}"'] \arrow[r, shift right, "\mathcal{L}"'] \arrow[r, phantom, "\perp"] & \mathbf{Dom} \arrow[l, shift right, "\mathcal{P}"']
\end{tikzcd}$$

The domain associated to a Petri net by the above construction can be obtained from that of any of its pre-net implementations by equating all the events which correspond to occurrences of the same transition with different linearizations of the same resources (which may differ for the order of tokens in the same place). Formally this is expressed as a natural transformation between the two semantics:

Theorem 4. *There is a natural transformation* $\varsigma : \mathcal{L} \circ \mathcal{E}_p \circ \mathcal{U}_p \to \mathcal{L} \circ \mathcal{E} \circ \mathcal{F} \circ \mathcal{U}_d \circ \mathcal{A}$.

As a consequence (as it happens for the algebraic models of computation) the domains associated to the pre-net implementations of a given net are all isomorphic, i.e., for all R, R', if $\mathcal{A}(R) \simeq \mathcal{A}(R')$ then $\mathcal{L} \circ \mathcal{E}_p \circ \mathcal{U}_p(R) \simeq \mathcal{L} \circ \mathcal{E}_p \circ \mathcal{U}_p(R')$.

Unfortunately, in the case of pre-nets finding a left adjoint for the functor $\mathcal{E}_p$ appears to be quite problematic. Intuitively, the left adjoint should freely generate an occurrence pre-net from any PES in a way which guarantees the existence and uniqueness of a representation of PES-morphisms in $\mathbf{PreOcc}_*$. Places could be freely generated as for ordinary Petri nets, but then it would be impossible to fix a linear order on the pre- and post-sets of transitions in a "universal" way. Our conjecture is that $\mathcal{E}_p$ is not a right adjoint functor.

An idea which seems promising in view of a universal characterisation of the mentioned construction is to abandon the purely algebraic view of pre-nets, considering an alternative notion of pre-net morphism, based on a weaker condition which requires the preservation of pre- and post-sets of transitions only up to a permutation. The permutation should be explicitly mentioned in the morphism itself, i.e., a morphism $f : R \to R'$ would be enriched with a family of permutations $\{\omega_t^0, \omega_t^1\}_{t \in T}$ such that $\omega_t : f_s^{\otimes}(\zeta_i(t)) \to \zeta_i(f_t(t))$ for any transition t in R.

4 Reconciling the unfolding and algebraic semantics of pre-nets

The unfolding of a marked pre-net can be seen as a maximal nondeterministic process, representing all its possible computations. Hence it is natural to expect that a tight relationship can be established between the unfolding and the algebraic / process approach. In this section we show that the domain produced through the unfolding construction can be obtained, equivalently, by means of a functorial construction based on the model of computation. The correspondence holds at categorical level, namely the functor $\mathcal{L} \circ \mathcal{E}_p \circ \mathcal{U}_p$ (see Fig. 9) and the new functor based on the algebraic semantics are naturally isomorphic. This improves the analogous result existing for Petri nets [MMS96], which only holds at the object level.

Let $\langle R, u\rangle$ be a marked pre-net and consider the comma category $\langle u \downarrow \mathcal{PP}(R)\rangle$ (which, by Theorem 2, is isomorphic to $\langle u \downarrow \mathcal{Z}(R)\rangle$). Objects are concatenable processes of R with source in u, and an arrow exists from a process δ_1 to δ_2 if $\delta_2 = \delta_1;\delta$ for some process δ. It can be shown that $\langle u \downarrow \mathcal{PP}(R)\rangle$ is a preorder, i.e., in $\langle u \downarrow \mathcal{PP}(R)\rangle$ there is at most one arrow between any two objects. Let $\lesssim_R$ denote the corresponding preorder relation i.e., $\delta_1 \lesssim_R \delta_2$ if there exists δ such that $\delta_1;\delta = \delta_2$.

An alternative characterisation of $\lesssim_R$, enforces the intuitive idea that it is a generalisation of the prefix ordering over processes. First, we need to introduce the notion of left injection for concatenable processes.

Definition 15 (left injection). *Let $\delta_i : u \to v_i$ $(i \in \{1,2\})$ be two objects in $\langle u \downarrow \mathcal{PP}(R)\rangle$, with $\delta_i = \langle \sigma_i, \pi_i, \tau_i\rangle$. A* left injection *$\iota : \delta_1 \to \delta_2$ is a morphism of pre-nets $\iota : R_{\pi_1} \to R_{\pi_2}$ (where R_{π_i} is the pre-net underlying π_i), such that*

1. *ι preserves the ordering of minimal places, namely $\sigma_2 = \iota_s^{\otimes}(\sigma_1)$;*
2. *ι is* rigid *on transitions, namely for t_2' in R_{π_2} and t_1 in R_{π_1}, if $t_2' \leq \iota(t_1)$ then $t_2' = \iota(t_1')$ for some t_1' in R_{π_1} (the image of a lower set is a lower set).*

The name "injection" comes from the fact that any morphism ι between marked deterministic occurrence nets results to be injective on places and transitions. The word "left" is related to the fact that ι is required to preserve only the string of minimal places.

Lemma 1. *Let $\delta_i : u \to v_i$ $(i \in \{1,2\})$ be objects in $\langle m \downarrow \mathcal{PP}(R)\rangle$, with $\delta_i = \langle \sigma_i, \pi_i, \tau_i\rangle$. Then $\delta_1 \lesssim_R \delta_2$ iff there exists a left injection $\iota : \delta_1 \to \delta_2$.*

$$
\begin{tikzcd}
& \mathbf{SSMC}_* \arrow[rr, "\curlywedge"] & & \mathbf{PreOrd} \arrow[dr, "\mathsf{Idl}"] & \\
\mathbf{PreNet}_* \arrow[ur, "\mathcal{Z}"] \arrow[dr, "\mathcal{U}_p"'] & & \Downarrow \rho & & \mathbf{PreOrd} \\
& \mathbf{PreOcc}_* \arrow[r, "\mathcal{E}_p"'] & \mathbf{PES} \arrow[r, "\mathcal{L}"'] & \mathbf{Dom} \arrow[ur, "\subseteq"'] &
\end{tikzcd}
$$

Fig. 10. Reconciling the algebraic and unfolding semantics of pre-nets.

By exploiting the above characterisation and the fact that $\mathcal{U}_p$ is a right adjoint we can conclude that the ideal completion of the preorder $\langle u \downarrow \mathcal{PP}(R)\rangle$, denoted by $\mathsf{Idl}(\langle u \downarrow \mathcal{PP}(R)\rangle)$, is isomorphic to the domain $\mathcal{L}(\mathcal{E}_p(\mathcal{U}_p(R)))$ obtained from the unfolding of the pre-net R.

To gain some intuition observe that the elements of the partial order induced by the preorder $\langle u \downarrow \mathcal{PP}(R)\rangle$ are classes of concatenable processes which are "left isomorphic", i.e., isomorphic via a left injection. Intuitively, the partial order consists of processes starting from a fixed initial state and ordered by prefix. Since processes are finite, taking the ideal completion of the partial order induced by the preorder $\langle u \downarrow \mathcal{PP}(R)\rangle$ (which produces the same result as taking directly the ideal completion of $\langle u \downarrow \mathcal{PP}(R)\rangle$) is necessary for moving from finite computations to arbitrary ones.

Theorem 5 (unfolding vs. concatenable processes). *Let $\langle R, u\rangle$ be a marked pre-net. Then $\mathsf{Idl}(\langle u \downarrow \mathcal{PP}(R)\rangle)$ is isomorphic to the domain $\mathcal{L}(\mathcal{E}_p(\mathcal{U}_p(R)))$.*

The above results admits a nice categorical formulation, since all the involved constructions can be seen as functors. Let **PreOrd** be the category of preorders and monotone functions, and let $\mathsf{Flat} : \mathbf{Cat} \to \mathbf{PreOrd}$ be the functor mapping any category to the underlying preorder (where $x \leq y$ if and only if there was an arrow $f : x \to y$ in the original category). Let $\curlywedge : \mathbf{SSMC}_* \to \mathbf{PreOrd}$ be the functor mapping any pointed symmetric strict monoidal category $\langle \mathbf{C}, O_{\mathbf{C}}\rangle$ to $\mathsf{Flat}(\langle O_{\mathbf{C}} \downarrow \mathbf{C}\rangle)$. Finally let $\mathbf{PreOrd} \to \mathbf{PreOrd}$ be the ideal completion functor, mapping any preorder to its ideal completion. Then the following result holds (see Fig. 10).

Theorem 6. *There is a natural isomorphism* $\rho : \mathsf{Idl} \circ \curlywedge \circ \mathcal{PP}_* \to \mathcal{L} \circ \mathcal{E}_p \circ \mathcal{U}_p$.

5 Adding read arcs

Several extensions of ordinary Petri nets have been proposed in the literature to enrich the expressiveness of the basic model. A mild generalisation which has been shown to be quite useful is the addition of the so-called *read arcs* which allow a transition to check for the presence of a token in a place without removing the token itself. Observe that a read arc cannot be safely replaced by a self-loop, since the former allows a greater amount of concurrency in the system: a resource can be read in parallel by several transitions at the same time, concurrently. For instance consider again the net N_0 in Fig. 1, and compare it to the net N_1 in Fig. 11, where place c is connected to transitions t_0 and t_1

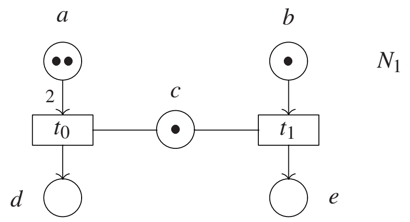

Fig. 11. Ordinary nets do not allow for concurrent read-only operations.

by read arcs (denoted by undirected lines), meaning that c represent a resource accessed in a read-only manner. While in N_1 the transitions t_0 and t_1 can fire concurrently, in the net N_0 where read arcs are replaced by self-loops, the two transitions are serialised.

Formally a *contextual Petri net* is a tuple $N = \langle \partial_0, \partial_1, \partial_2, S, T \rangle$, where $\langle \partial_0, \partial_1, S, T \rangle$ is an ordinary Petri net and $\partial_2 : T \to S^{\oplus}$ associates to each transition its context. Notice that, as a single token can be read concurrently by different transitions, it can be read also with multiplicity greater than 1 by the same transition. Hence a transition t can use, to fire, any marking ranging from $\partial_0(t) \oplus [\![\partial_2(t)]\!]$ to $\partial_0(t) \oplus \partial_2(t)$, i.e., it is sufficient that any context place contains at least one token. The notion of *marked contextual net* and of (marked) contextual net morphism are defined as structure-preserving graph homomorphisms in the obvious way, yielding the categories **CPetri** and **CPetri**$_*$.

Correspondingly, we consider an extension of pre-nets with read arcs.

Definition 16 (contextual pre-net). *A contextual pre-net is a tuple $R = \langle \zeta_0, \zeta_1, \zeta_2, S, T \rangle$ such that $\langle \zeta_0, \zeta_1, S, T \rangle$ is a pre-net and $\zeta_2 : T \to S^{\otimes}$ is the* context *function.*

The notion of pre-net morphism can be extended to contextual pre-nets in the obvious way, requiring for any transition the preservation of the context, besides pre- and post-conditions. Also the extension to marked pre-nets is immediate. We denote by **CPreNet** and by **CPreNet**$_*$ the corresponding categories.

The algebraic semantics of contextual pre-nets has been developed in [BMMS02] taking as models the so-called *match-share categories*, a kind of symmetric monoidal category equipped with two additional (non-natural) transformations. Let $\mathcal{Z}_c(R)$ denote the model of computation for a contextual pre-net R, as defined in such paper (as a special case, in the absence of contexts, $\mathcal{Z}_c(R) = \mathcal{Z}(R)$). The construction can be expressed as an adjunction and it is defined in terms of theory morphisms between suitable equational theories. Due to space limitation, we cannot give full details here.

The results developed in this paper for Petri nets and pre-nets generalise to the contextual case. In the following we sketch the basic notions, constructions and the results involved in the extension.

First, as it happens for ordinary contextual nets [BCM01], the dependencies among events in a contextual pre-net computation cannot be captured completely by two binary relations representing causality and symmetric conflict. While *causality* can be defined essentially as in the ordinary case, due to the possibility of preserving part of the state in a step of computation, an *asymmetric* form of conflict arises between transitions. In fact let t, t' be transitions such that $\zeta_2(t) = s = \zeta_0(t')$. Then the firing of t' prevents t

$$\frac{1 \leq i \leq |u|}{u'_i = \langle \emptyset, u_i, i \rangle \in S' \quad \eta_s(u'_i) = u_i}$$

$$\frac{v_p, v_c \in S'^{\otimes} \quad v_p \text{ safe} \quad v_p \cap v_c = \emptyset \quad co([v_p \cdot v_c]) \quad t \in T \quad \eta_s^{\otimes}(v_p) = \zeta_0(t) \quad \eta_s^{\otimes}(v_c) = \zeta_2(t)}{t' = \langle v_p, v_c, t \rangle \in T' \quad \eta_t(t') = t \quad \zeta'_0(t') = v_p \quad \zeta'_2(t') = v_c}$$

$$\frac{t' = \langle v, t \rangle \in T' \quad \zeta_1(t) = w_1 \ldots w_n}{w'_i = \langle t', w_i, i \rangle \in S' \quad \eta_s(w'_i) = w_i \quad \zeta'_1(t') = w'_1 \ldots w'_n}$$

Fig. 12. Inference rules for the unfolding $\mathcal{U}_c(\langle R, u \rangle)$ of a contextual pre-net R.

to be fired, since it consumes the shared resource in s. Instead the firing of t just reads a resource in s and thus t' can fire after t. This kind of dependency is represented by introducing an *asymmetric conflict* relation $\nearrow$ on transitions, which models the previous situation as $t \nearrow t'$. An ordinary symmetric conflict, arising when two transition t and t' have a common precondition, is represented as an asymmetric conflict in both directions, i.e., $t \nearrow t' \nearrow t$. Finally, since $<$ represents a global order of execution, while $\nearrow$ determines an order of execution only locally to each computation, it is natural to impose $\nearrow$ to be an extension of $<$.

The notion of *concurrency* is updated to take into account the presence of asymmetric conflict: a set of places $X \subseteq S$ is *concurrent*, written $co(X)$, if for any $s, s' \in X$ it does not hold $s < s'$, $\lfloor X \rfloor$ is finite and $\nearrow$ is acyclic on $\lfloor X \rfloor$.

Then the contextual occurrence pre-nets can be naturally defined.

Definition 17 (occurrence contextual pre-net). *An* occurrence contextual pre-net *is a safe pre-net R such that (i) causality $<_R$ is a partial order; (ii) R has no backward conflicts; (iii) for any transition t, the set of causes $\lfloor t \rfloor$ is finite and asymmetric conflict $\nearrow_R$ is acyclic on $\lfloor t \rfloor$. An occurrence contextual pre-net is* deterministic *if it has no forward conflicts.*

We denote by **CPreOcc**$_*$ *full subcategory of* **CPreNet**$_*$ *having marked occurrence contextual pre-nets as objects.*

In the unfolding construction below just notice that the second rule takes a context v_c which is not required to be safe, consistently with the fact that single token can be read with multiplicity greater than 1.

Definition 18 (contextual unfolding). *Let $\langle R, u \rangle$ be a marked contextual pre-net. The unfolding $\mathcal{U}_c(\langle R, u \rangle) = ((\zeta'_0, \zeta'_1, \zeta'_2, S', T'), u')$ and the* folding morphism $\eta_R = \langle \eta_t, \eta_s \rangle :$ *$\mathcal{U}_c(R) \to R$ are the occurrence pre-net and (elementary) contextual pre-net morphism inductively defined by the rules in Fig. 12, with $u' = \langle \emptyset, u_1, 1 \rangle \ldots \langle \emptyset, u_{|u|}, |u| \rangle$.*

Also in this case the unfolding extends to a functor $\mathcal{U}_c$: **CPreNet**$_* \to$ **CPreOcc**$_*$ which is right adjoint to the inclusion of **CPreOcc**$_*$ into **CPreNet**$_*$. The unfolding can be abstracted to an event based model, called *asymmetric event structure* (AES's), introduced in [BCM01] as a generalisation of Winskel PES's where conflict is allowed to be non-symmetric. As proved in the mentioned paper, the category of AES's coreflects into **Dom** allowing to recover a domain semantics. The situation is summarised in Fig. 13.

$$\begin{tikzcd}
\mathbf{CPreNet}_* \arrow[r, bend right, "\mathcal{U}_c"'] \arrow[r, phantom, "\perp"] & \mathbf{CPreOcc}_* \arrow[l, hook', bend right] \arrow[r, "\mathcal{E}_c"] & \mathbf{AES} \arrow[r, bend right, "\mathcal{L}"'] \arrow[r, phantom, "\perp"] & \mathbf{Dom} \arrow[l, bend right, "\mathcal{P}"']
\end{tikzcd}$$

Fig. 13. Denotational semantics of contextual pre-nets.

	Algebraic	Process	Logical	Unfolding	Reconciliation
P/T nets CTPh	**[MM90]**	**[BD87]**	**[BMMS01]**		
P/T nets ITPh	*[DMM96,Sas98]*	*[GR83,DMM96,Sas98]*		**[Win87,MMS97a]**	*[MMS96]*
pre-nets ITPh	**[BMMS01]**	**Section 2**	**[BMMS01]**	Section 3	Section 4

Fig. 14. Net semantics.

The algebraic and unfolding approach to the semantics of contextual pre-nets can be reconciled, along the same schema followed for pre-net, obtaining a commutative functorial diagram which generalises Fig. 10 in the presence of read arcs.

6 Conclusions

We have shown that a functorial unfolding semantics for pre-nets can be developed along the lines of the seminal work of Winskel. The semantics is expressed as a chain of functors leading from the category $\mathbf{PreNet}_*$ to the category **Dom**, through $\mathbf{PreOcc}_*$ and **PES**. A different construction of a domain for any pre-net can be defined by relying on the algebraic semantics of pre-nets, already defined in the literature. Differently from what happens for Petri nets, this latter construction can be expressed as a functor from $\mathbf{PreNet}_*$ to **Dom**. The unfolding and algebraic constructions can be reconciled in a fully satisfactory categorical setting, by showing that the corresponding functors are naturally isomorphic. The proof relies on the introduction of a concrete notion of process for pre-nets, and on a characterisation of the algebraic semantics in terms such processes.

Figure 14 summarises our results, connecting them to the known (CTPh and ITPh) net semantics. Each column is devoted to a specific semantic flavour (see the classification in the Introduction). The last column refers to the possibility of relating the algebraic and unfolding views. The entries are either references to the literature where the corresponding construction has been presented, or pointers to the sections of our contribution. Empty cells stands for unfeasible constructions. Italic text refers to non-functorial constructions, i.e., constructions that are defined just at the object level, but cannot deal with simulation morphisms. Regular entries stands for functorial constructions, and bold entries for adjunctions. Note that, in the case of pre-nets, all constructions are feasible and functorial. Finally, we mention that all constructions and results for pre-nets are extended to work in the presence of read arcs.

References

[BCM01] P. Baldan, A. Corradini, and U. Montanari. Contextual Petri nets, asymmetric event structures and processes. *Inform. and Comput.*, 1(171):1–49, 2001.

[BD87] E. Best and R. Devillers. Sequential and concurrent behaviour in Petri net theory. *Theoret. Comput. Sci.*, 55:87–136, 1987.

[Ber78] G. Berry. Stable models of typed lambda-calculi. In *Proceedings of ICALP'78*, vol. 62 of *Lect. Notes in Comput. Sci.*, pages 72–89. Springer-Verlag, 1978.

[BMMS99] R. Bruni, J. Meseguer, U. Montanari, and V. Sassone. Functorial semantics for Petri nets under the individual token philosophy. In *Proceedings of CTCS'99*, vol. 29 of *Elect. Notes in Th. Comput. Sci.* Elsevier Science, 1999.

[BMMS01] R. Bruni, J. Meseguer, U. Montanari, and V. Sassone. Functorial models for Petri nets. *Inform. and Comput.*, 170(2):207–236, 2001.

[BMMS02] R. Bruni, J. Meseguer, U. Montanari, and V. Sassone. Functorial models for contextual pre-nets. Technical Report TR-02-09, University of Pisa, 2002.

[CW01] F. Crazzolara and G. Winskel. Events in security protocols. In *Proceedings of CCS'01*, pages 96–105. ACM, 2001.

[DFMR94] N. De Francesco, U. Montanari, and G. Ristori. Modeling concurrent accesses to shared data via Petri nets. In *Programming Concepts, Methods and Calculi*, vol. A-56 of *IFIP Transactions*, pages 403–422. North Holland, 1994.

[DMM96] P. Degano, J. Meseguer, and U. Montanari. Axiomatizing the algebra of net computations and processes. *Acta Inform.*, 33(7):641–667, 1996.

[GP95] R.J. van Glabbeek and G.D. Plotkin. Configuration structures. In *Proceedings of LICS'95*, pages 199–209. IEEE Computer Society Press, 1995.

[GR83] U. Goltz and W. Reisig. The non-sequential behaviour of Petri nets. *Inform. and Comput.*, 57:125–147, 1983.

[Mac71] S. MacLane. *Categories for the Working Mathematician*. Springer, 1971.

[MM90] J. Meseguer and U. Montanari. Petri nets are monoids. *Inform. and Comput.*, 88:105–155, 1990.

[MMS92] J. Meseguer, U. Montanari, and V. Sassone. On the semantics of Petri nets. In *Proceedings of CONCUR '92*, vol. 630 of *Lect. Notes in Comput. Sci.*, pages 286–301. Springer-Verlag, 1992.

[MMS96] J. Meseguer, U. Montanari, and V. Sassone. Process versus unfolding semantics for Place/Transition Petri nets. *Theoret. Comput. Sci.*, 153(1-2):171–210, 1996.

[MMS97a] J. Meseguer, U. Montanari, and V. Sassone. On the semantics of Place/Transition Petri nets. *Math. Struct. in Comput. Sci.*, 7:359–397, 1997.

[MMS97b] J. Meseguer, U. Montanari, and V. Sassone. Representation theorems for Petri nets. In *Foundations of Computer Science: Potential - Theory - Cognition*, vol. 1337 of *Lect. Notes in Comput. Sci.*, pages 239–249. Springer, 1997.

[MR94] U. Montanari and F. Rossi. Contextual occurrence nets and concurrent constraint programming. In *Graph Transformations in Computer Science*, vol. 776 of *LNCS*, pages 280–295. Springer, 1994.

[MR95] U. Montanari and F. Rossi. Contextual nets. *Acta Inform.*, 32:545–596, 1995.

[NPW81] M. Nielsen, G. Plotkin, and G. Winskel. Petri Nets, Event Structures and Domains, Part 1. *Theoret. Comput. Sci.*, 13:85–108, 1981.

[Pet62] C.A. Petri. *Kommunikation mit Automaten*. PhD thesis, Schriften des Institutes für Instrumentelle Matematik, Bonn, 1962.

[Rei85] W. Reisig. *Petri Nets: An Introduction*. EATCS Monographs on Theoretical Computer Science. Springer, 1985.

[Sas98] V. Sassone. An axiomatization of the category of Petri net computations. *Math. Struct. in Comput. Sci.*, 8(2):117–151, 1998.

[Vog97] W. Vogler. Efficiency of asynchronous systems and read arcs in Petri nets. In *Proceeding of ICALP'97*, vol. 1256 of *LNCS*, pages 538–548. Springer, 1997.

[Win87] G. Winskel. Event Structures. In *Petri Nets: Applications and Relationships to Other Models of Concurrency*, vol. 255 of *LNCS*, pages 325–392. Springer, 1987.