Acta Informatica
https://doi.org/10.1007/s00236-018-0314-0

ORIGINAL ARTICLE

Petri nets are dioids: a new algebraic foundation for non-deterministic net theory

Paolo Baldan[1] · Fabio Gadducci[2]

Received: 6 September 2016 / Accepted: 14 January 2018

Abstract In a seminal paper Montanari and Meseguer have shown that an algebraic interpretation of Petri nets in terms of commutative monoids can be used to provide an elegant characterisation of the deterministic computations of a net, accounting for their sequential and parallel composition. A smoother and more complete theory for deterministic computations has been later developed by relying on the concept of pre-net, a variation of Petri nets with a non-commutative flavor. This paper shows that, along the same lines, by adding an (idempotent) operation and thus considering dioids (idempotent semirings) rather than just monoids, one can faithfully characterise the non-deterministic computations of a net.

1 Introduction

Petri nets [30] are one of the most studied and best known models for concurrent systems. Due to the conceptual simplicity of the model and its intuitive graphical presentation, since their introduction, which dates back to the Sixties [29], they have attracted the interest of both theoreticians and practitioners.

The basic operational behaviour of Petri nets can be straightforwardly defined in terms of sequences of transition firings, according to the "token game". Concurrency in computations can be made explicit by resorting to a semantics given in terms of *(non-sequential) deterministic processes* à la Goltz and Reisig [15]. A process describes the transition firings in a computation and their mutual dependency relations. Concretely, a deterministic processes is an acyclic, deterministic net whose structure induces a partial order on transitions, which can be seen as occurrences of transition firings in the original net. A deterministic process

✉ Fabio Gadducci
fabio.gadducci@unipi.it

Paolo Baldan
baldan@math.unipd.it

1 Dipartimento di Matematica, Università di Padova, Via Trieste 63, 35121 Padua, Italy

2 Dipartimento di Informatica, Università di Pisa, Largo Bruno Pontecorvo 3, 56127 Pisa, Italy

Published online: 24 January 2018

captures an abstract notion of concurrent computation, in the sense that it can be seen as a representative of a full class of firing sequences differing only for the order of independent firings, i.e., all the firing sequences corresponding to linearisations of the underlying partial ordering.

The concurrent nature of Petri net computations has been expressed in an elegant way in the "Petri nets are monoids" approach [26]. Processes can be naturally seen as arrows of a symmetric monoidal category, called the *computational model* of the net, where arrow composition is sequential composition on processes and the monoidal operation is parallel composition. The crux is that such computational model can be characterised as the symmetric monoidal category $\mathscr{P}(N)$ freely generated from N, in the same way as the same way as the cartesian category $\mathscr{L}(\Sigma)$ of terms and substitutions is freely generated from a signature Σ. Processes of N are arrows of $\mathscr{P}(N)$ and the functoriality of the monoidal operator $\otimes$ capture the essence of concurrency in net computations. Since $\mathscr{P}(N)$ can be finitely axiomatised, one also gets an algebraic characterisation of deterministic processes (see also the summary in [24]).

A smoother theory has been later developed relying on *pre-nets* [7], a variant of ordinary nets where a total ordering is imposed on the places in the pre- and post-sets of transitions. Using strings rather than multisets allows one to uniquely characterise an element by its position. The algebraic presentation of the model of computation based on pre-nets turns out to be simpler and more satisfactory as the construction yields an adjunction between the category of pre-nets and the category of models (i.e., of symmetric monoidal categories [23]). Since any pre-net can be seen as a concrete "implementation" of the Petri net obtained by forgetting the ordering of places in pre- and post-sets and any two such implementations generate isomorphic categories of processes, this yields also an unequivocal construction for ordinary Petri nets.

Besides concurrency, a crucial aspect in the behaviour of a distributed system is the non-deterministic interaction between its components. Indeed, most process calculi provide parallel composition and non-deterministic sum as basic structuring operators. For Petri nets, non-determinism naturally arises when distinct transitions compete for the same resources (see e.g. [31], and [1] or the recent book [16] for a survey on net encodings of process calculi). At the level of Petri net processes, different deterministic computations from the same starting state can be merged into a single non-deterministic process [12,28], a structure that, besides concurrency, also captures the intrinsic non-determinism of a net. A non-deterministic process is again a net satisfying suitable acyclicity requirement, but where, roughly speaking, transitions can compete for the use of tokens, thus leading to a branching structure.

Even if non-deterministic Petri net processes have been widely investigated and used for verification purposes (see e.g. [13]), to the best of our knowledge they have always been formalised set-theoretically and no formal connection has been drawn from a categorical model to a set-theoretical notion of non-deterministic process.

In this paper we show that the "Petri nets are monoids" approach can be naturally generalised in order to capture the non-deterministic computations of a Petri net. In brief, we prove that an algebraic model of the non-deterministic computations of Petri nets can be obtained by resorting to what we call *diodal categories*, where, besides the monoidal operator $\otimes$ for parallel composition, a second monoidal operator $\oplus$ is used for modelling the non-deterministic composition of computations.

The presence of two symmetric monoidal operators $\otimes$ and $\oplus$, the former distributing over the latter, naturally leads to consider *bimonoidal* (or *rig*) categories. Roughly speaking, they are a categorical counterpart of semirings (or rigs) pretty much as monoidal categories corresponds to monoids, and as such they have surfaced in the literature on distributed systems (e.g. in the specification of flowchart schemes [35]). The branching structure of non-deterministic computations further requires the presence of a natural transformation $\nabla_a : a \to a \oplus a$. Roughly, it allows for the duplication of a resource a into the non-deterministic choice between two copies of a, which can be used in conflicting, non-deterministic computations. Naturality enforces the property that the non-deterministic choice between two copies of a computation is the same as a single copy of such computation, exposing non-deterministically two copies of the target state. For instance, in the simple case of a computation consisting of a single transition $t : u \to v$, we have that $\nabla_u; t : u \to v \oplus v$ is the same as $t; \nabla_v$. The fact that this recalls the idempotency axioms of $\oplus$ in tropical semirings or dioids, led us to refer the corresponding categorical structure as a dioidal category.

More technically, as in the deterministic case, we focus on pre-nets and since different pre-nets corresponding to the same Petri net have isomorphic models of computation, this gives indirectly a construction for Petri nets. For any pre-net R we introduce the corresponding category of concatenable non-deterministic processes $\mathbf{NP}(R)$. Then we show that the category $\mathbf{NP}(R)$ of concatenable non-deterministic processes can be characterised as the free dioidal category $\mathbf{ANP}(R)$ built over R. As a consequence, the non-deterministic processes of a Petri net N, as introduced in [12], turn out to be in a one-to-one correspondence with the arrows of $\mathbf{ANP}(R)$, for any pre-net R implementing N, quotiented under suitable axioms.

The paper is organised as follows. In Sect. 2, after recalling some basics about Petri nets and pre-nets, we review the notions of deterministic and non-deterministic process. In Sect. 3 we present the construction of the category of concatenable non-deterministic processes for a Petri net. In Sect. 4 we recall some basic notions on symmetric monoidal categories, we introduce dioidal categories and we present the main theorem, concerning the correspondence between non-deterministic processes and arrows of a free dioidal category. Finally, in Sect. 5 we highlight the connection of our work with PROP theory and we provide a topological interpretation of our algebraic construction of non-deterministic processes. The paper is rounded up with some further categorical remarks on our proposal and some pointers to further works.

The paper is a follow-up to [3]. A relevant difference concerns the use of pre-nets with respect to Petri nets, and their interpretation in terms of graph theory. This required a large rewriting of the visual and set-theoretical presentation of Petri nets, yet ensuring a smoother axiomatisation of the overall algebraic structure, which takes advantage of the current research on the graphical representation of PROPs (product and permutation categories) [5,19,21].

2 Petri nets and non-deterministic processes

Given a set X, we denote by X^* the free semigroup over X (finite non-empty strings of elements of X) and by $X^{\circledast}$ the free commutative semigroup over X (finite non-empty multisets over X). We write $\mu : X^* \to X^{\circledast}$ for the function mapping a string to the underlying multiset. Moreover, given a function $f : X \to Y^*$ we denote by $f^* : X^* \to Y^*$ its obvious semigroup extension. Similarly, given a function $f : X \to Y^{\circledast}$ we denote by $f^{\circledast} : X^{\circledast} \to Y^{\circledast}$ its commutative semigroup extension. Given $u \in X^*$ or $u \in X^{\circledast}$, we denote by $[\![u]\!]$ the underlying subset of X defined in the obvious way, e.g., for $u \in X^*$, $[\![u]\!] = \{x \in X \mid \exists u', u'' \in X^{\circledast}.\ u = u'xu''\}$. When set relations are used over strings or multisets, we implicitly refer to the underlying set. E.g., for $u, v \in X^*$ (or in $X^{\circledast}$) by $x \in u$ we mean $x \in [\![u]\!]$ and by $u \cap v$ we mean $[\![u]\!] \cap [\![v]\!]$.

Definition 1 (*P/T net*) A *P/T Petri net* is a 4-tuple $N = (\delta_0, \delta_1, S, T)$, where S is a non-empty set of *places*, T is a set of *transitions*, and $\delta_0, \delta_1 : T \to S^{\circledast}$ are functions assigning multisets called pre- and post-set, respectively, to each transition.

Hereafter, for any net N we assume $N = (\delta_0, \delta_1, S, T)$, with subscripts and superscripts carrying over the names of the components. Observe that, since the pre- and post-set of a transition are elements of $S^{\circledast}$, they are necessarily non-empty. This requirement, which is quite common in Petri net theory, allows us to simplify the presentation, but the theory could be adapted to deal with transitions having possibly empty pre- and post-sets. The issue is discussed in Sect. 5.

Given a net N, a multiset $u \in S^{\circledast}$, representing a state of the net, is referred to as a *marking* of N. It is *safe* if any place occurs at most once in it, i.e., $u = [\![u]\!]$.

The operational behaviour is expressed in terms of the "token game". A transition t is *enabled* by a marking m if its pre-set is "covered" by m. In this case, t can be fired: this consumes the tokens in the pre-set and produces those in the post-set. Formally, the firing rule is

$$\delta_0(t) \circledast u \to_t \delta_1(t) \circledast u$$

for $u \in S^{\circledast}$ and $t \in T$.

The notion of net morphism naturally arises from an algebraic view, where places and transitions play the role of sorts and operators.

Definition 2 (*Net morphism*) A *Petri net morphism* $f = \langle f_{\mathrm{s}}, f_{\mathrm{t}} \rangle : N \to N'$ is a pair where $f_{\mathrm{s}} : S^{\circledast} \to S'^{\circledast}$ is a (commutative) semigroup homomorphism and $f_{\mathrm{t}} : T \to T'$ is a function such that $\delta_i' \circ f_{\mathrm{t}} = f_{\mathrm{s}} \circ \delta_i$, for any $i \in \{0, 1\}$. The category of P/T nets (as objects) and their morphisms (as arrows) is denoted by **Petri**.

In the sequel, when the meaning is clear from the context, we omit the subscripts from the morphism components, thus writing f instead of f_{s} and f_{t}.

2.1 Pre-nets and their morphisms

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

Definition 3 (*Pre-net*) A *pre-net* is a 4-tuple $R = (\zeta_0, \zeta_1, S, T)$, where S is a non-empty set of *places*, T is a set of *transitions*, and $\zeta_0, \zeta_1 : T \to S^*$ are functions assigning strings called pre- and post-set, respectively, to each transition.

For a pre-net R we will denote by S_R and T_R its sets of places and transitions.

The pictorial representation of Petri nets played an important role in their large diffusion as a specification framework. This graphical presentation (places represented as circles, transitions as boxes, pre- and post-set multirelations as weighted arcs, tokens as black bullets) is extended to pre-nets by adopting the following convention: weighted arcs are replaced by arcs labelled with the *ordered list of positions* in which the place appears in either the pre- or the post-set of the transition. An example of pre-net R_0 is depicted in the right part of Fig. 1, and it is going to be used throughout the paper to illustrate definitions and concepts. The arc from a to t is decorated with the annotation 1, 3, which means that the firing of t requires two tokens from a, to be taken as first and third consumed resources, while the second token

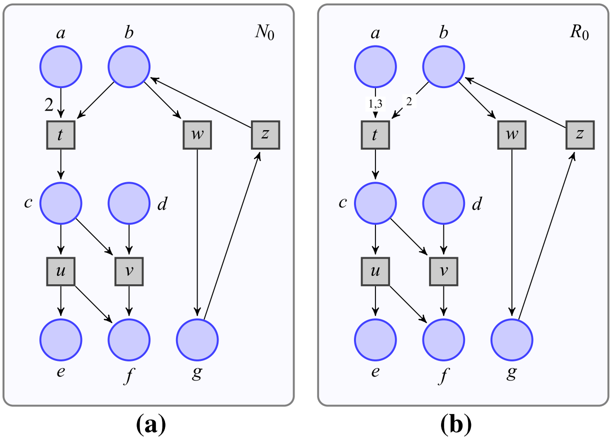


Fig. 1 **a** The P/T Petri net N_0 and **b** one of its pre-net implementation R_0

to be consumed by t must be taken from b, as expressed by annotation 2 of the arc from b to t (we stress that for pre-nets 2 denotes a position, not the number of tokens to be consumed). The decoration is omitted when the pre-set or the post-set of a transition includes just a single place. For instance, for transition t it is intended that $\zeta_0(t) = aba$ and $\zeta_1(t) = c$.

Also pre-nets, when viewed as algebraic structures, can be naturally endowed with a notion of morphism.

Definition 4 (*Pre-net morphism*) A *pre-net morphism* is a pair $f = \langle f_\mathrm{s}, f_\mathrm{t}\rangle : R \to R'$ where $f_\mathrm{s} : S^* \to S'^*$ is a semigroup homomorphism, and $f_\mathrm{t} : T \to T'$ is a function such that $\zeta_i' \circ f_\mathrm{t} = f_\mathrm{s} \circ \zeta_i$, for $i \in \{0, 1\}$. The category of pre-nets (as objects) and their morphisms (as arrows) is denoted by **Pre**.

Pre-nets can thus 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 that is obtained from R by replacing any string with the corresponding multiset. This construction is formalised as a functor $\mathscr{A}$: **Pre** $\to$ **Petri** defined as

- any pre-net $R = (\zeta_0, \zeta_1, S, T)$ is mapped to $\mathscr{A}(R) = (\delta_0, \delta_1, S, T)$, where $\delta_i(t) = \mu(\zeta_i(t))$ for each $t \in T$ and $i \in \{0, 1\}$;
- any pre-net morphism $f : R \to R'$ is mapped to $\mathscr{A}(f) = \langle g_\mathrm{s}^{\circledast}, f_\mathrm{t}\rangle$, where $g_\mathrm{s}(s) = \mu(f_\mathrm{s}(s))$ for each $s \in S$.

For instance, referring to Fig. 1, the ordinary Petri net N_0 on the left is implemented by the pre-net R_0 on the right, i.e., we have $\mathscr{A}(R_0) = N_0$. The transition $t : 2a \circledast b \to c$ in N_0 is implemented as $t : aba \to c$ in R_0 and $u : c \to e \circledast f$ in N_0 as $u : c \to fe$ in R_0. Alternative implementations are possible via different linearisations.

A computation of a pre-net consists, intuitively, of a sequence of "explicit" steps, firings of transitions that consume and produce resources, and of "implicit" steps, which rearrange the order of the resources to allow for the application of transitions. The operational rules are

- $u * \zeta_0(t) * u' \to_t u * \zeta_1(t) * u'$ and
- $u * a * b * u' \to u * b * a * u'$

for $u, u' \in S^*$, $t \in T$, and $a, b \in S$.

All the sequences of implicit steps that implement the same permutation of a given state are indistinguishable. This will be formalised later in the paper for deterministic and non-deterministic computations. It can be shown that given a pre-net R and two states $u, v \in S^*$, the state v is reachable from u if and only if $\mu(v)$ is reachable from $\mu(u)$ in $\mathscr{A}(R)$ [7]. For this reason, abusing the terminology, we sometime informally say that in a pre-net R a marking m' is reachable from a marking m, meaning that from $u \in S^*$ such that $\mu(s) = m$ we can reach u' such that $\mu(u') = m'$.

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 deterministic models of computation [7]. As we are going to see, the same happens in the non-deterministic case, hence from now on we will focus mainly on the semantics of pre-nets.

2.2 Occurrence pre-nets

Let R be a pre-net. The *causality relation* is the least transitive relation $<_R \subseteq (S \cup T) \times (S \cup T)$ such that

- if $s \in \zeta_0(t)$ then $s <_R t$;
- if $s \in \zeta_1(t)$ then $t <_R s$.

Given a place or transition $x \in S \cup T$, the set of *causes* of x in T is defined as $\lfloor x \rfloor = \{t \in T \mid t <_R x\} \cup \{x\}$; and, for $X \subseteq S \cup T$, $\lfloor X \rfloor = \bigcup_{x \in X} \lfloor x \rfloor$. The *conflict relation* $\#_R \subseteq (S \cup T) \times (S \cup T)$ is the least symmetric relation such that

i. if $t \neq t'$ and $\zeta_0(t) \cap \zeta_0(t') \neq \emptyset$ then $t \#_R t'$;
ii. if $x \#_R x'$ and $x' <_R x''$ then $x \#_R x''$.

Definition 5 (*Occurrence pre-net*) An *occurrence pre-net* is a pre-net O where $\mu(\zeta_o(t))$, $\mu(\zeta_1(t))$ are safe for all $t \in T_O$ and (i) causality $<_O$ is a strict partial order; (ii) there are no backward conflicts, i.e., $\zeta_1(t) \cap \zeta_1(t') = \emptyset$ for any $t \neq t'$; (iii) conflict $\#_O$ is irreflexive. The sets of minimal and maximal places of O with respect to $<_O$ are denoted by $\min(O)$ and $\max(O)$. An occurrence pre-net is *deterministic* if it has no forward conflicts, i.e., for any $t \neq t'$, $\zeta_0(t) \cap \zeta_0(t') = \emptyset$.

An occurrence pre-net O can be seen as the representation of a possibly non-deterministic computation starting from (some sequence of places corresponding to) $\min(O)$. Since each place can occur at most once in the pre- and post-set of a transition of an occurrence pre-net, in the graphical representation the labelling of the arcs is sometimes omitted, meaning that the pre- or post-set should be read from left to right. For instance, for the occurrence pre-net in the left part of Fig. 2, $\zeta_0(t_1) = a_1 b a_2$, similarly $\zeta_0(t_2) = a_1 b a_2$, while $\zeta_0(v_1) = c_1 d_1$. Reachable states in O can be characterised statically by using the dependency relations, via the notion of cut.

Definition 6 (*Cuts*) Let O be an occurrence pre-net. A *cut* in O is a maximal (with respect to set inclusion) subset of places $X \subseteq S_O$ such that neither $s <_O s'$ nor $s \#_O s'$ for all $s, s' \in X$. The set of cuts of O is denoted by $cuts(O)$. A subset of cuts $W \subseteq cuts(O)$ is called a *covering* of O if $\max(O) \subseteq \bigcup_{X \in W} X$.

It can be shown that any cut $X \in cuts(O)$ is reachable from $\min(O)$ by executing all the transitions in $\lfloor X \rfloor$ in any order compatible with $<_O$ (see, e.g., [36]). The notion of covering is original to this paper: intuitively, if a subset W of cuts is intended to represent the set of final states of a set of computations of O, then W is a covering for O if each maximal place of

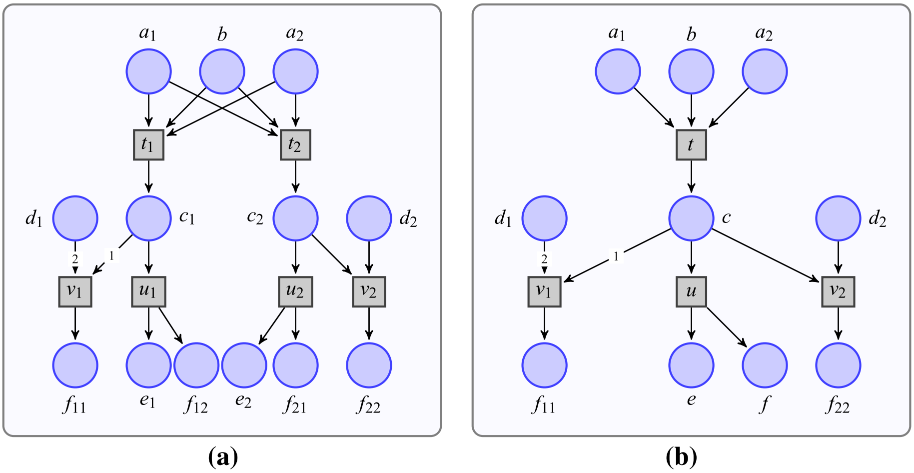


Fig. 2 **a** A redundant process $\xi : O \to N$ and **b** its collapsing $\pi(\xi) : O' \to N$

O is eventually filled in at least one of those computations. It is easy to see that the covering condition amounts to ask $T_O = \bigcup_{X \in W} \lfloor X \rfloor$, i.e., each transition of O is used in at least one of the computations reaching the states in W.

2.3 Processes

We now review the notion of deterministic and non-deterministic process for pre-nets. A process is represented as a morphism $\pi : O \to R$ from an occurrence pre-net O to the original pre-net R [15]. Since morphisms are simulations, they map computations of O into computations of R in such a way that a process can be seen as a representative of a set of possible computations of R. The occurrence pre-net makes explicit the causal structure of such computations since each transition is fired at most once and each place is filled with at most one token during each computation. Transitions and places of O can be thought of, respectively, as firing of transitions and tokens in places of the original net. Actually, to allow for such an interpretation, further restrictions have to be imposed on the morphism π, namely it must map places into places (rather than into strings of places).

Definition 7 (*Process*) Let R be a pre-net. A *non-deterministic redundant process* of R is a pre-net morphism $\pi : O \to R'$ where O is a finite occurrence pre-net and for all $s \in S$, $f_s(s) \in S'$. We call $\pi : O \to R$ a *non-deterministic process* when it further satisfies that if $\pi(t) = \pi(t')$ and $\zeta_0(t) = \zeta_0(t')$ then $t = t'$ for any $t, t' \in T_O$ (irredundancy).

The process π is *deterministic* if the underlying occurrence pre-net O is so. For a process π we write $\min(\pi)$, $\max(\pi)$ and $cuts(\pi)$ to refer to the sets $\min(O)$, $\max(O)$ and $cuts(O)$ in the underlying occurrence pre-net.

Intuitively, a process π represents a set of possible computations starting at some $u \in {S_R}^*$, determined by a chosen ordering on the minimal places $\min(\pi)$ of π. The irredundancy condition is motivated by the common assumption that the non-deterministic composition of a computation with itself should give back the original computation. Requiring irredundancy prevents to have different processes representation of the same set of concurrent computations [12]. Hereafter, consistently with the literature, all non-deterministic processes will be implicitly assumed to be irredundant. The qualification "irredundant" will be omitted and, only when referring to processes not satisfying the irredundancy condition, we will explicitly qualify them as redundant processes and denote them with the letter ξ, possibly with subscripts. Some hints at the theory arising in the absence of the irredundancy assumption and the graphical interpretation of (ir)redundancy are further discussed in Sect. 5.

We next observe that, if we fix an ordering on the minimal places of a process π, namely we view them as a string, we can identify a canonical form for π along the lines of what is done in [12] for ordinary Petri nets.

Definition 8 (*Canonical process*) Let R be a pre-net, $\pi : O \rightarrow R$ a non-deterministic process such that $T_O \subseteq T_R \times {S_O}^*$ and $\alpha \in {S_O}^*$ a string such that $\mu(\alpha) = \min(\pi)$. We call π *α-canonical* if

- $\alpha = \langle s_1, 1\rangle \ldots \langle s_n, n\rangle$ with $s_i \in S_R$ and $\pi(\langle s_i, i\rangle) = s_i$;
- $\pi(\hat{t}) = t$, $\zeta_0(\hat{t}) = u$, and $\zeta_1(\hat{t}) = \langle s_1, \hat{t}\rangle \ldots \langle s_m, \hat{t}\rangle$ with $s_j \in S_R$ and $\pi(\langle s_j, \hat{t}\rangle) = s_j$ for $\hat{t} = \langle t, u\rangle \in T_O$.

Observe that, by the definition above, we have that $\pi(x)$ is the projection on the first component for any $x \in S_O \cup T_O$ and that $\zeta_0(t)$ is the projection on the second component for any $t \in T_O$.

Given the inductive nature of the notion of canonicity, for any process $\pi : O \rightarrow R$ and string $\alpha \in {S_O}^*$ such that $\mu(\alpha) = \min(\pi)$ we can uniquely define a canonical process $\pi' : O' \rightarrow R$ and an isomorphism $f : O \rightarrow O'$ such that π' is $f^*(\alpha)$-canonical.

More generally, any redundant process $\xi : O \rightarrow R$ can be turned into a canonical process. The point is that, once a source state is fixed, the process is uniquely determined (up to isomorphism) and it is obtained via a construction that builds a canonical form for the process. Observe, in fact, that a canonical process implicitly satisfies the irredundancy condition since, because of the naming scheme, it prevents to have two transitions with the same pre-set and same image in the original net.

Proposition 1 (Collapsing) *Let R be a pre-net, $\xi : O \rightarrow R$ a redundant process and $\alpha \in {S_O}^*$ a string such that $\mu(\alpha) = \min(O)$. Then there exists a unique factorisation $\xi = \beta; \pi(\xi)$, where β is epi and $\pi(\xi)$ is a $\beta^*(\alpha)$-canonical process. The process $\pi(\xi)$ is called the α-*collapsing *of ξ.*

Proof Assume $\min(O) = \{s_1, \ldots, s_n\}$ with $\alpha = s_1 \ldots s_n$. The canonical process $\pi(\xi)$ is determined by the place and transition sets of the underlying occurrence pre-net O'. We define such sets and the morphism β, inductively, as follows

- for any $i \in \{1, \ldots, n\}$, let $s'_i = \langle \xi(s_i), i\rangle \in S_{O'}$ and $\beta(s_i) = s'_i$;
- for any $t \in T_O$, let
  - $t' = \langle \xi(t), \beta^*(\zeta_0(t))\rangle \in T_{O'}$;
  - $\zeta_0(t') = \beta^*(\zeta_0(t))$;
  - if $\zeta_1(t) = s_1 \ldots s_k$ then let $s'_j = \langle \xi(s_j), t'\rangle \in S_{O'}$ for $j \in \{1, \ldots, k\}$, and define $\zeta_1(t') = s'_1 \ldots s'_k$.
  - $\beta(t) = t'$ and $\beta(s_i) = s'_i$ for $j \in \{1, \ldots, k\}$.

It is immediate to see that β and $\pi(\xi)$ are well-defined and that $\xi = \beta; \pi(\xi)$. □

The occurrence pre-net in Fig. 2a represents a redundant process of the pre-net R_0 in Fig. 1b. The α-collapsing of such a process, for $\alpha = d_1 a_1 b a_2 d_2$, is depicted in Fig. 2b. The morphisms are implicitly represented by naming places and transitions of the occurrence pre-net as their images over R_0, possibly with subscripts. For the sake of readability, items are not named according to Definition 8. For instance, places denoted d_1, a_1, b, a_2, and d_2 in $\pi(\xi)$ are actually $\langle d, 1\rangle$, $\langle a, 2\rangle$, $\langle b, 3\rangle$, $\langle a, 4\rangle$, and $\langle d, 5\rangle$, respectively. Transition t is $\langle t, \langle a, 2\rangle\langle b, 3\rangle\langle a, 4\rangle\rangle$, and so on.

Given two α-canonical processes π_1 and π_2 for a pre-net R, it is immediate to see that their point-wise union $\pi_1 \cup \pi_2$ is again a α-canonical process. More generally, mimicking [12], one can show that the set of α-canonical processes ordered by sub-set inclusion is a lattice, with union and intersection as join and meet.

We finally prove that each process can be obtained as the union of its deterministic sub-processes. In order to formalise this fact, first note that given an occurrence pre-net O, any causally closed subset of transitions $T' \subseteq T_O$ (i.e., such that if $t \in T_O$ then $\lfloor t \rfloor \subseteq T_O$) induces a non-deterministic sub-net O' of O, with set of transitions T' and set of places $S' = \min(O) \cup \bigcup_{t\in T'}([\![\zeta_0(t)]\!] \cup [\![\zeta_1(t)]\!])$.

Definition 9 (*Sub-processes*) Let R be a pre-net, $\pi : O \to R$ a process and $X \in cuts(\pi)$ a cut. The projection of π on the cut X, denoted $\pi \downarrow X$, is the deterministic process $\pi' : O' \to R$, where O' is the sub-net of O induced by $\lfloor X \rfloor$ and π' is the corresponding restriction of π.

It is immediate to see that $\pi \downarrow X$ is indeed deterministic (since its transitions $\lfloor X \rfloor$ cannot include conflicts by the definition of cut).

Proposition 2 (Process decomposition) *Let R be a pre-net, $\pi : O \to R$ a α-canonical process for some α, and $W \subseteq cuts(\pi)$. Then $\pi \downarrow X$ is α-canonical for any $X \in W$. Moreover, if W is a covering, then $\pi = \bigcup_{X\in W} \pi \downarrow X$.*

Proof Just observe that by construction the set of transitions of the occurrence pre-net O' underlying the process $\bigcup_{X\in W} \pi \downarrow X$ is $T' = \bigcup_{X\in W} \lfloor X \rfloor$.

If W is a covering of π, then T' is the entire set of transitions T_O of the occurrence pre-net underlying π and therefore $O' = O$. Since a canonical process is completely determined by the underlying occurrence pre-net, we conclude. □

By exploiting the result above, we show that we can perform on a process an operation that is a sort of converse of collapsing, meaning that it maximises redundancy.

Definition 10 (*Maximally redundant process*) Let R be a pre-net. A redundant process $\xi : O \to R$ is called *maximally redundant* whenever irredundancy may fail only on minimal places, i.e., if $\xi(t) = \xi(t')$ and $\zeta_0(t) = \zeta_0(t')$ then either $\zeta_0(t) \subseteq \min(\xi)$ or $t = t'$ for any $t, t' \in T_O$.

By Proposition 2, given a process $\pi : O \to R$ and a covering $W \subseteq cuts(\pi)$, we can build a corresponding maximally redundant (canonical) process $\xi : O' \to R$, whose collapsing is π. If $\min(\pi) = \{s_1, \ldots, s_k\}$ and $W = \{X_1, \ldots, X_n\}$, just take the disjoint union $\bigcup_{i=1}^{n}(\pi \downarrow X_i) \times \{i\}$ (where all operations are taken componentwise), quotiented under the equivalence that equates different instances of the same minimal place, namely $\langle s_i, h'\rangle \sim \langle s_i, h''\rangle$, for $i \in \{1, \ldots, k\}$ and $h', h'' \in \{1, \ldots, n\}$.

Going back to Fig. 2, the process ξ on the left is actually a maximally redundant process for the pre-net R_0 in Fig. 1b, and indeed, it is the maximally redundant canonical process associated to the process $\pi(\xi)$ on the right.

3 Concatenable processes

In this section, after reviewing the theory of concatenable deterministic process of pre-nets from [7] (generalising [11,32]), we propose a notion of concatenable non-deterministic process. This leads to a category $\mathbf{NP}(R)$ of non-deterministic processes for a pre-net R, where objects are states and arrows model non-deterministic computations of R. In turn, this gives a category for the corresponding Petri net.

3.1 Concatenable deterministic processes

Concatenable deterministic processes for Petri nets have been introduced [11,32] as a refinement of Goltz-Reisig deterministic processes, endowed with operations of sequential and parallel composition that are consistent with the causal structure of computations. In order to properly define such operations, for ordinary nets, a suitable ordering have to be imposed over the places in $\min(\pi)$ and $\max(\pi)$ for each process π. Such an ordering allows to distinguish among "interface" places of O_π that are mapped to the same place of the original net, a capability that is essential to track causal dependencies. For pre-nets this is built-in in the notion of state itself.

Definition 11 (*Concatenable deterministic process*) A *concatenable deterministic process* δ of a pre-net R is a triple $\langle \alpha, \pi, \omega \rangle$, where π is a deterministic process of R and $\alpha, \omega \in {S_O}^*$ are such that

$$\mu(\alpha) = \min(\pi) \quad \text{and} \quad \mu(\omega) = \max(\pi).$$

We denote by $\zeta_0(\delta)$ the string $\pi^*(\alpha)$ and by $\zeta_1(\delta)$ the string $\pi^*(\omega)$.

Given two concatenable deterministic processes $\langle \alpha, \pi, \omega \rangle$ and $\langle \alpha', \pi', \omega' \rangle$, an isomorphism between them is an isomorphism of the underlying pre-nets $f : O \to O'$, consistent with the mapping to the original pre-net and with the linearisations of minimal and maximal places, i.e., $\pi' \circ f = \pi$, $f^*(\alpha) = \alpha'$, and $f^*(\omega) = \omega'$. The isomorphism class of a concatenable deterministic process δ is written $[\delta]$ and called an *abstract* concatenable deterministic process. Often the word "abstract" is omitted and δ is used to denote the corresponding isomorphism class.

Concatenable deterministic processes $\langle \alpha, \pi, \omega \rangle$ of pre-nets are graphically represented as follows: (1) places and transitions are named by their images through π, with subscripts; (2) the source α and target ω are represented explicitly at the top and bottom of the process, respectively; (3) pre- and post-set of transitions are to be read from left to right, unless arcs are decorated for giving a different order. The representation is one-to-one, meaning that isomorphic processes are mapped to the same representation up-to graph isomorphism. See Sect. 5 for a discussion.

In Fig. 3 we report some deterministic processes for the pre-net R_0 of Fig. 1b that correspond to single transitions. Also basic processes corresponding to place identities and permutations are shown. The latters play a role in computations: as observed before, since states are strings of places, sometimes a reordering can be necessary for concatenating two computations (Fig. 4).

Definition 12 (*Sequential and parallel composition*) Let $\delta_1 = \langle \alpha_1, \pi_1, \omega_1 \rangle$ and $\delta_2 = \langle \alpha_2, \pi_2, \omega_2 \rangle$ be two concatenable deterministic processes of a pre-net N.

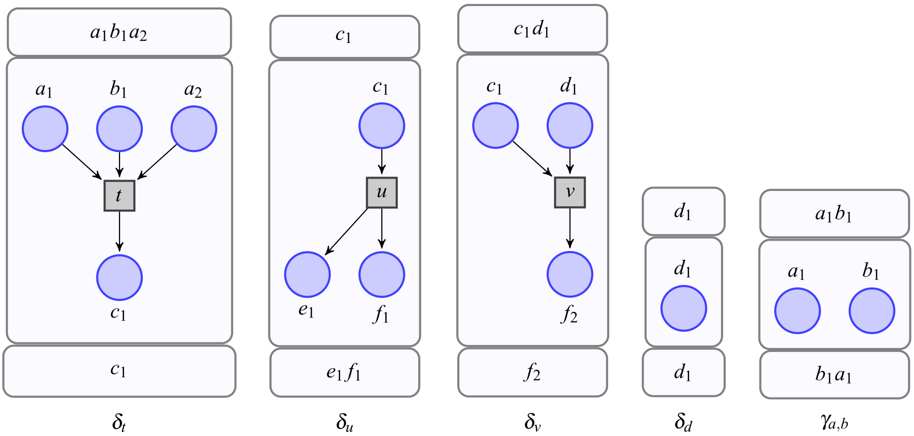


Fig. 3 Graphical representation of simple deterministic processes

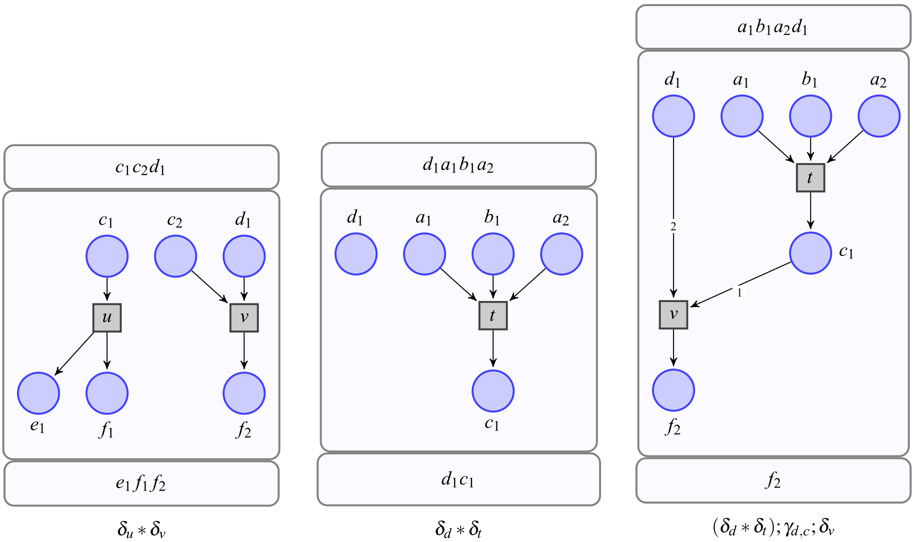


Fig. 4 Tensor product and sequential composition of simple deterministic processes

- Let $\zeta_1(\delta_1) = \zeta_0(\delta_2)$ and assume $T_1 \cap T_2 = \emptyset$ and $S_1 \cap S_2 = \max(\pi_1) = \min(\pi_2)$, with $\omega_1 = \alpha_2$. Then $\delta_1; \delta_2$ is the concatenable deterministic process $\langle \alpha_1, \pi, \omega_2 \rangle$, where process π is the (component-wise) union of π_1 and π_2.
- Assume $T_1 \cap T_2 = S_1 \cap S_2 = \emptyset$. Then $\delta_1 * \delta_2$ is the concatenable process $\langle \alpha_1\alpha_2, \pi, \omega_1\omega_2 \rangle$, where process π is the (component-wise) union of π_1 and π_2.

The premise of the first item means that δ_1 and δ_2 overlap only on $\max(\pi_1) = \min(\pi_2)$, and on such places the labelling on the original pre-net and the ordering coincide. Then, their concatenation is the process obtained by gluing the maximal places of π_1 and the minimal places of π_2 according to their ordering. Parallel composition is obtained by juxtaposing the two processes: under the assumption that the underlying pre-nets have disjoint sets of places and transitions, this reduces to point-wise union.

Concatenation and parallel composition, as it can be routinely checked, induce well-defined operations on abstract deterministic processes, which are independent of the choice of representatives.

Definition 13 (*Category of concatenable deterministic processes*) Let R be a pre-net. The *category of (abstract) concatenable deterministic processes* of R, denoted by $\mathbf{P}(R)$, is defined as follows: objects are non-empty strings of places of R, i.e. elements of S^*; each (abstract) concatenable process $[\delta]$ of R is an arrow from $\zeta_0(\delta)$ to $\zeta_1(\delta)$.

We will discuss later how parallel composition actually induces the structure of a monoidal category on $\mathbf{P}(R)$.

3.2 Concatenable non-deterministic processes

Intuitively, a concatenable non-deterministic process is a set of non-deterministic processes that, starting from a set of possible initial states, produces a set of possible final states. For technical reasons, analogous to those which suggested to move from nets to pre-nets, it is preferable to consider sequences, rather than sets, of processes.

Definition 14 (*Concatenable non-deterministic process*) Let R be a pre-net. A *concatenable non-deterministic process* η for R is a triple of non-empty lists $\langle \boldsymbol{\alpha}, \boldsymbol{\pi}, \boldsymbol{\omega} \rangle$ with

- $\boldsymbol{\pi} = \pi_1 \ldots \pi_n$ is a list of (pair-wise disjoint) non-deterministic processes;
- $\boldsymbol{\alpha} = \alpha_1 \ldots \alpha_n$ is a list of strings such that $\alpha_i \in S_{O_i}{}^*$ and $\mu(\alpha_i) = \min(\pi_i)$;
- $\boldsymbol{\omega} = \omega_1 \ldots \omega_\ell$ is a list of strings such that
  - $\omega_j \in S_{O_i}{}^*$ for some $i \in \{1, \ldots, n\}$ and $\mu(\omega_j) \in \mathit{cuts}(\pi_i)$;
  - $\bigcup_{i=1}^{n} \max(\pi_i) \subseteq \bigcup_{j=1}^{l} \mu(\omega_j)$, i.e., for any $i \in \{1, \ldots, n\}$ the cuts of π_i occurring in $\mu^*(\boldsymbol{\omega})$ are a covering of π_i.

The source of η is the list $\zeta_0(\eta) = \pi_1{}^*(\alpha_1) \ldots \pi_n{}^*(\alpha_n)$, i.e., the list of the sources of the component processes, while the target of η is $\zeta_1(\eta) = u_1 \ldots u_\ell$, where $u_j = \pi_i{}^*(\omega_j)$ if $\mu(\omega_j) \in \mathit{cuts}(\pi_i)$.

The covering requirement guarantees that all the maximal places of each underlying deterministic process appear on the target, thus ensuring that all transitions are actually executed in some computation and the produced state is visible in the output. A weakening of this condition, which allows to discard (and thus hide) the result of some deterministic sub-computation, is discussed in Sect. 5.

In order to ease notation we fix the above naming scheme: unless stated otherwise, we assume concatenable non-deterministic processes to be of the kind $\langle \boldsymbol{\alpha}, \boldsymbol{\pi}, \boldsymbol{\omega} \rangle$, with $\boldsymbol{\alpha} = \alpha_1 \ldots \alpha_n$, $\boldsymbol{\pi} = \pi_1 \ldots \pi_n$, and $\boldsymbol{\omega} = \omega_1 \ldots \omega_l$. In turn, for each process π_i in $\boldsymbol{\pi}$ we assume $\pi_i : O_i \to N$, where O_i has S_i and T_i as place and transition sets, respectively.

Two concatenable non-deterministic processes $\langle \boldsymbol{\alpha}, \boldsymbol{\pi}, \boldsymbol{\omega} \rangle$ and $\langle \boldsymbol{\alpha'}, \boldsymbol{\pi}', \boldsymbol{\omega'} \rangle$ are *isomorphic* if $|\boldsymbol{\pi}| = |\boldsymbol{\pi'}|$ and there exist non-deterministic process isomorphisms $f_i : \pi_i \to \pi'_i$, with $i \in \{1, \ldots, |\boldsymbol{\pi}|\}$, consistent with the decorations and the ordering of sources and targets, i.e., such that, for all $i \in \{1, \ldots, n\}$, $f_i^*(\alpha_i) = \alpha'_i$ and for all $j \in \{1, \ldots, \ell\}$, if $\mu(\omega_j) \in \mathit{cuts}(\pi_i)$, then $f_i^*(\omega_j) = \omega'_j$.

The canonical form for processes can easily be transferred to concatenable processes, thus getting a choice for the canonical representative in an isomorphism class.

Definition 15 (*Canonical concatenable processes*) Let R be a pre-net. A concatenable process $\langle \boldsymbol{\alpha}, \boldsymbol{\pi}, \boldsymbol{\omega} \rangle$ of R is *canonical* if for any $i \in \{1, \ldots, |\boldsymbol{\pi}|\}$ the process π_i is α_i-canonical.

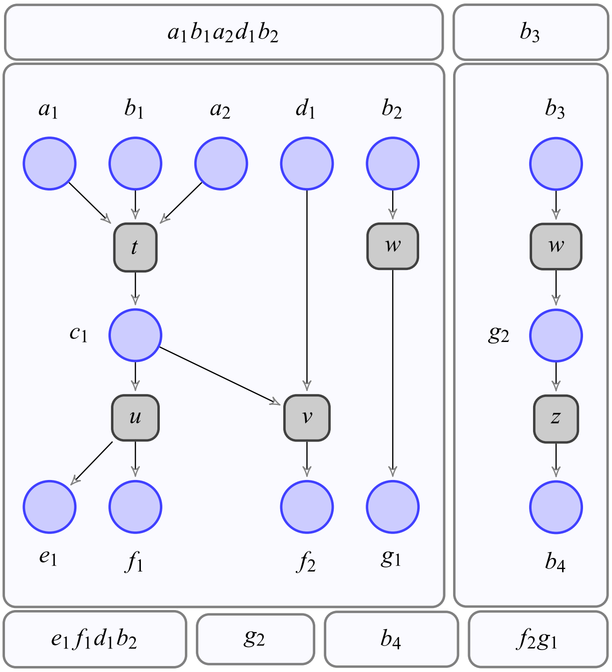


Fig. 5 A concatenable non-deterministic process

It is easy to show that two concatenable non-deterministic processes are isomorphic if and only if they have the same canonical representation. Thus, *abstract* concatenable non-deterministic processes, i.e., isomorphism classes of processes, are often identified with their canonical representatives: we write η to refer to the corresponding abstract process.

Graphically, a concatenable non-deterministic process $\langle \boldsymbol{\alpha}, \boldsymbol{\pi}, \boldsymbol{\omega} \rangle$ is represented by a list of occurrence pre-nets $O_1 \dots O_n$, underlying the component sub-processes, each one enclosed in a separate box. Places and transitions of O_i are named by their images through π_i, with subscripts. The source α_i of each process π_i is represented at the top of the process itself, while in the bottom part of the box we represent $\boldsymbol{\omega}$ as a list of boxes, one for each ω_j. As an example, a concatenable process for the pre-net R_0 in Fig. 1b is reported in Fig. 5. The process is $\langle \alpha_1\alpha_2, \pi_1\pi_2, \omega_1\omega_2\omega_3\omega_4 \rangle$ consisting of two component processes π_1 and π_2 with four targets. Concerning the targets, $\{e_1, f_1, d_1, b_2\}, \{f_2, g_1\} \in cuts(\pi_1)$ and $\{g_2\}, \{b_4\} \in cuts(\pi_2)$. It is easy to see that the cuts $\{e_1, f_1, d_1, b_2\}, \{f_2, g_1\}$ are a covering of O_1, and similarly $\{g_2\}, \{b_4\}$ are a covering for O_2. As for deterministic processes, there is a one-to-one correspondence between canonical non-deterministic processes and their graphical representation.

Sequential and non-deterministic composition for concatenable non-deterministic processes of a given pre-net R can be defined as follows.

Definition 16 (*Sequential composition*) Let $\eta = \langle \boldsymbol{\alpha}, \boldsymbol{\pi}, \boldsymbol{\omega} \rangle$ and $\eta' = \langle \boldsymbol{\alpha}', \boldsymbol{\pi}', \boldsymbol{\omega}' \rangle$ be concatenable non-deterministic processes of a pre-net R such that $\zeta_1(\eta) = \zeta_0(\eta')$ (thus $|\boldsymbol{\omega}| = |\boldsymbol{\alpha}'|$). Assume, for any i, j, that $T_i \cap T'_j = \emptyset$ and if $\mu(\omega_j) \in cuts(\pi_i)$ then $S_i \cap S'_j = \mu(\omega_j) = \min(\pi'_j)$, with $\omega_j = \alpha'_j$. We define $\eta_1; \eta_2 = \langle \boldsymbol{\alpha}'', \boldsymbol{\pi}'', \boldsymbol{\omega}'' \rangle$, where

- $\boldsymbol{\pi}'' = \pi''_1 \dots \pi''_{|\boldsymbol{\pi}|}$ and each process π''_i is obtained as follows: take the (component-wise) union of π_i with all processes $\pi'_1, \dots, \pi'_h$ such that $\mu(\omega_j) \in cuts(\pi_i)$ for $j \in \{1, \dots, h\}$,

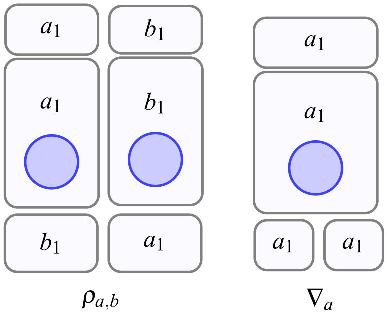


Fig. 6 Basic non-deterministic processes

thus getting a redundant process $\xi_i : O''_i \to R$; consider the α_i-collapsing $\beta_i ; \pi(\xi_i)$ of such morphism and let $\pi''_i = \pi(\xi_i)$;

– $\boldsymbol{\alpha}'' = \beta_1{}^*(\alpha_1) \dots \beta_{|\boldsymbol{\pi}|}{}^*(\alpha_{|\boldsymbol{\pi}|})$ and $\boldsymbol{\omega}'' = \beta_1{}^*(\omega'_1) \dots \beta_{|\boldsymbol{\omega}'|}{}^*(\omega'_{|\boldsymbol{\omega}'|})$.

Roughly, for any $j \in \{1, \dots, |\boldsymbol{\omega}|\}$, if $\mu(\omega_j)$ is a cut in π_i, then the process π'_j in η' must be attached to the set of places $\mu(\omega_j)$ in π_i. Assuming that π_i and π'_j overlap only on $\min(\pi'_j)$, and that on such places the labelling on the original pre-net and the ordering imposed by the two processes coincide, attaching π_j to π_i reduces to taking their component-wise union. Thus the composition has $|\boldsymbol{\pi}|$ components, where the i-th component is obtained as the component-wise union of π_i with all the π'_j that must be connected to π_i.

We can easily define also a notion of non-deterministic composition, which is obtained by juxtaposing the two processes.

Definition 17 (*Non-deterministic composition*) Let $\eta = \langle \boldsymbol{\alpha}, \boldsymbol{\pi}, \boldsymbol{\omega} \rangle$ and $\eta' = \langle \boldsymbol{\alpha}', \boldsymbol{\pi}', \boldsymbol{\omega}' \rangle$ be concatenable non-deterministic processes of a pre-net R. Assume $T_i \cap T'_j = S_i \cap S'_j = \emptyset$ for any i, j. Then $\eta \oplus \eta' = \langle \boldsymbol{\alpha\alpha}', \boldsymbol{\pi\pi}', \boldsymbol{\omega\omega}' \rangle$, where the juxtaposition of two lists denotes their concatenation.

Differently from parallel composition, note that the two component processes $\boldsymbol{\pi}$ and $\boldsymbol{\pi}'$ are kept separate in a way that $\eta \oplus \eta'$ represents the non-deterministic choice between the execution of $\boldsymbol{\pi}$ from $\boldsymbol{\alpha}$ and the execution of $\boldsymbol{\pi}'$ from $\boldsymbol{\alpha}'$.

It is easy to prove that non-deterministic and parallel composition induce well-defined operations on abstract processes, independent of the choice of representatives.

Some concatenable non-deterministic processes that does not involve transitions are used to perform some "re-organisation" of the resources. In particular, since the non-deterministic components are ordered in a list, their sequential composition may require a rearrangement as expressed by the permutation process $\rho_{a,b}$ in Fig. 6(left). Moreover, the same target can be exposed non-deterministically in two copies, as in the duplicator process ∇_a in Fig. 6(right). In Fig. 7 we show the composition $\delta_u \oplus \delta_v$ and the processes $(\delta_t * \delta_d)$; ∇_{c*d}; $[(\delta_u * \delta_d) \oplus \delta_v]$ and δ_t; δ_u; ∇_{e*f}, which are obtained by using the processes in Fig. 3.

Definition 18 (*Category of concatenable non-deterministic processes*) Let R be a pre-net. The *category of (abstract) concatenable non-deterministic processes* of R, denoted by $\mathbf{NP}(R)$, is defined as follows: objects are elements of $S^{\langle * \rangle}$, i.e., non-empty lists of elements of S^*; each (abstract) concatenable non-deterministic process of R is an arrow.

The non-deterministic processes of a pre-net R, given in Definition 7, correspond to the arrows of $\mathbf{NP}(R)$ consisting of a list of processes of length one, i.e., of the kind $\langle \alpha_1, \pi_1, \omega \rangle$,

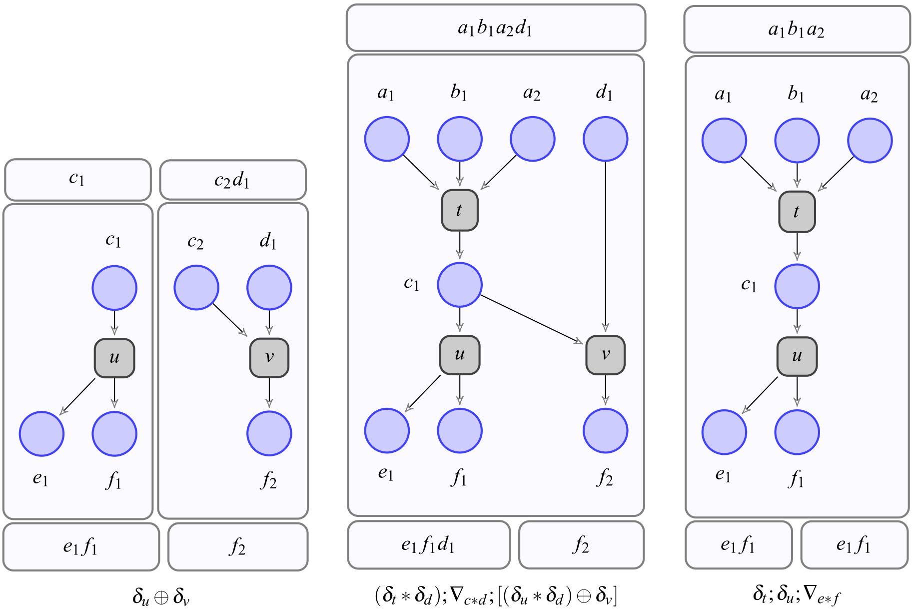


Fig. 7 Some non-deterministic processes arising from the processes in Fig. 3

once we forget the decoration. An immediate adaptation of a result for deterministic processes [7, Theorem 3.1] shows that different pre-net implementations of the same Petri net have isomorphic categories of non-deterministic processes.

Proposition 3 *Let R and R' be pre-nets such that $\mathscr{A}(R) = \mathscr{A}(R')$. Then $\mathbf{NP}(R) = \mathbf{NP}(R')$.*

3.3 A decomposition theorem

Relying on the canonical form for concatenable processes we can provide a decomposition theorem for their non-deterministic counterpart. As a first step, we introduce a notation for processes representing general forms of permutations and duplications. In the sequel, given an equivalence $\sim$ and a permutation ϕ on $[1, k]$ we say that ϕ is *stable* with respect to $\sim$ if it preserves the relative order of elements in the same class, i.e., for $i, j \in [1, k]$, $i \leq j$ and $i \sim j$ implies $\phi(i) \leq \phi(j)$.

Definition 19 (*Permutation and duplicator processes*) Let R be a pre-net. Given $u_1, \ldots, u_k \in S^*$ and a permutation $\phi : [1, k] \to [1, k]$, we write $\rho^{\phi}_{u_1,\ldots,u_k} : u_1 \oplus \cdots \oplus u_k \to u_{\phi(1)} \oplus \cdots \oplus u_{\phi(k)}$ for the process offering as target the ϕ-permutation of the source. We say that $\rho^{\phi}_{u_1,\ldots,u_k}$ is *stable* with respect to an equivalence $\sim$ on $[1, k]$ if ϕ is stable. Given $u \in S^*$ and an integer $k \geq 1$, we denote by $\nabla^k_u : u \to u \oplus \cdots \oplus u$ the concatenable non-deterministic process with source u and target k copies of u.

As an example, if $\phi : [1, 2] \to [1, 2]$ is the permutation defined by $\phi(1) = 2$ and $\phi(2) = 1$, then $\rho^{\phi}_{a,b}$ is the process $\rho_{a,b}$ in the left part of Fig. 6.

Lemma 1 (Decomposition for concatenable non-deterministic processes) *Let R be a pre-net and $\eta = \langle \boldsymbol{\alpha}, \boldsymbol{\pi}, \boldsymbol{\omega} \rangle$ a concatenable process of R.*

1. *If* $|\boldsymbol{\pi}| = 1$ *then* η *can be decomposed uniquely as*

$$\nabla^{|\boldsymbol{\omega}|}_{\alpha_1}; (\eta_1 \oplus \cdots \oplus \eta_{|\boldsymbol{\omega}|})$$

   with $\eta_j = \langle \alpha_1, \pi_1 \downarrow \mu(\omega_j), \omega_j \rangle$*, where* $j \in \{1, \ldots, |\boldsymbol{\omega}|\}$.
2. *If* $|\boldsymbol{\pi}| > 1$ *then* η *can be decomposed as*

$$\eta = (\eta_1 \oplus \cdots \oplus \eta_{|\boldsymbol{\pi}|}); \rho$$

   with $\eta_i = \langle \alpha_i, \pi_i, \omega_{i_1} \ldots \omega_{i_j} \rangle$*, where* $\omega_{i_1}, \ldots, \omega_{i_j}$ *are the components of* $\boldsymbol{\omega}$ *such that* $\mu(\omega_{i_h}) \in cuts(\pi_i)$ *and* $i_1 \leq i_2 \leq \cdots \leq i_j$.
   The decomposition is unique if we additionally require that ρ *does not alter the relative order of the targets of the* η_i*, i.e., if* ρ *is stable with respect to the equivalence* $\Sigma_{j=0}^{h-1} i_j + k_1 \sim \Sigma_{j=0}^{h-1} i_j + k_2$*, for* $h \in [1, |\boldsymbol{\pi}|]$ *and* $k_1, k_2 \in [1, i_h]$.

Proof 1. We can assume, without loss of generality, that π_1 is α_1-canonical. By Proposition 2, each process $\pi_1 \downarrow \mu(\omega_j)$ is α_1-canonical, namely η_j is canonical for $j \in \{1, \ldots, |\boldsymbol{\omega}|\}$. According to Definitions 16 and 17, $\nabla^{|\boldsymbol{\omega}|}_{\alpha_1}; (\eta_1 \oplus \cdots \oplus \eta_{|\boldsymbol{\omega}|})$ is obtained by taking the disjoint union of $\eta_1, \ldots, \eta_{|\boldsymbol{\omega}|}$, keeping only ω in common, and then considering the collapsing. It is immediate to see that this corresponds to take the pointwise union of the η_j, which by Proposition 2 gives η, as desired.
Concerning uniqueness, suppose that there is another decomposition of the same shape $\nabla^{|\boldsymbol{\omega}|}_{\alpha_1}; (\eta'_1 \oplus \cdots \oplus \eta'_{|\boldsymbol{\omega}|})$. If we assume that each η'_j is canonical, then $\eta = \eta'_1 \cup \cdots \cup \eta'_{|\boldsymbol{\omega}|}$. However, since each η'_j is uniquely determined by the cut, then we have that $\eta_j = \eta'_j$ for $j = \{1, \ldots, |\boldsymbol{\omega}|\}$.
2. Just note that $\eta_1 \oplus \cdots \oplus \eta_{|\boldsymbol{\pi}|}$, according to Definition 17, is obtained by taking the disjoint union of the η_j's and thus it has the same source and same underlying process as η. Instead, in the target we have first the components ω_j of the target of η which are in η_1, then those which are in η_2 and so on. This means that the target of $\eta_1 \oplus \cdots \oplus \eta_{|\boldsymbol{\pi}|}$ is a permutation of the target of η, whence the thesis.
Uniqueness follows by observing that the target of each η_i is uniquely determined by the request of preserving the order of the strings in $\boldsymbol{\omega}$, and also the symmetry ρ is determined by the requirement of stability. □

In the second case each process can be further decomposed in the normal form stated in the first case, thus also the combined decomposition is unique. The construction of the normal form mimics, at the categorical level, the construction of a maximally redundant process corresponding to the given process, as described at the end of Sect. 2.3: we further elaborate on this in the following sections, when providing our algebraic characterisation for (possibly redundant) non-deterministic processes.

4 Embedding processes into terms

This section presents the core result of the paper, namely, the description of the abstract concatenable non-deterministic processes of a pre-net R, as defined in Sect. 3, in terms of a suitable algebra. Along the *Petri nets are monoids* paradigm, this is a sort of monoidal category, freely generated from the pre-net itself.

4.1 Categorical notions

Here we introduce the relevant categorical notions that are needed for the algebraic description of processes. Most definitions are standard: for the presentation of monoidal categories we closely follow [6], while for PROPs and bipermutative categories we refer to [19] and [25], respectively.

Definition 20 (*Permutative categories*) A (strict) *monoidal category* is a pair $\langle \mathscr{C}, \_ \otimes \_ \rangle$, where $\mathscr{C}$ is the underlying category, the *tensor product* $\_ \otimes \_ : \mathscr{C} \times \mathscr{C} \longrightarrow \mathscr{C}$ is a functor such that the objects $(a \otimes b) \otimes c$ and $a \otimes (b \otimes c)$ coincide, and satisfying the coherence axioms $\iota_{a \otimes b} = \iota_a \otimes \iota_b$ (for ι_a the identity of object a) and

$$\begin{tikzcd}
a\otimes(b\otimes c) \arrow[r, "="] \arrow[d, "t_1\otimes(t_2\otimes t_3)"'] & (a\otimes b)\otimes c \arrow[d, "(t_1\otimes t_2)\otimes t_3"] \\
a_1\otimes(b_1\otimes c_1) \arrow[r, "="'] & (a_1\otimes b_1)\otimes c_1
\end{tikzcd}$$

A *permutative category* is a triple $\langle \mathscr{C}, \_ \otimes \_, \gamma \rangle$, where $\langle \mathscr{C}, \_ \otimes \_ \rangle$ is a monoidal category and $\gamma : \_{}_1 \otimes \_{}_2 \Rightarrow \_{}_2 \otimes \_{}_1 : \mathscr{C} \times \mathscr{C} \longrightarrow \mathscr{C}$ is a natural isomorphism[1] satisfying the coherence axioms below

$$\begin{tikzcd}
a\otimes b\otimes c \arrow[r, "\iota_a\otimes\gamma_{b,c}"] \arrow[dr, "\gamma_{a\otimes b,c}"'] & a\otimes c\otimes b \arrow[d, "\gamma_{a,c}\otimes\iota_b"] \\
& c\otimes a\otimes b
\end{tikzcd}
\qquad
\begin{tikzcd}
a\otimes b \arrow[r, "\gamma_{a,b}"] \arrow[dr, "\iota_a\otimes\iota_b"'] & b\otimes a \arrow[d, "\gamma_{b,a}"] \\
& a\otimes b
\end{tikzcd}$$

Since the monoidal operation $\otimes$ of permutative categories is associative on both objects and arrows, from now on we will drop parenthesis. Often, they are also referred to as *symmetric (strict) monoidal categories*. However, in order to ease the presentation, with respect to the standard definitions we removed the requirement of a unity object. Hence, with an abuse of terminology, we call a monoidal category what should be called a semigroup category. We will return on this issue in Sect. 5.

There is a tight connection with PROP categories, as witnessed by the definition below (see [19, Remark 16]).

Definition 21 (*PROPs*) A PROP is a 4-tuple $\langle S, \mathscr{C}, \_ \otimes \_, \gamma \rangle$, where $\langle \mathscr{C}, \_ \otimes \_, \gamma \rangle$ is a permutative category such that the objects of $\mathscr{C}$ are elements of S^*, and $\otimes$ acts as string concatenation on objects, i.e., $s_1 \otimes s_2 = s_1 s_2$.

We now enrich the set of arrows, in order to consider idempotent operators.

Definition 22 (*G-monoidal categories*) A *g-monoidal category* is a 4-tuple $\langle \mathscr{C}, \_ \oplus \_, \rho, \nabla \rangle$, where $\langle \mathscr{C}, \_ \oplus \_, \rho \rangle$ is a permutative category and $\nabla : \_{}_1 \Rightarrow \_{}_1 \oplus \_{}_1 : \mathscr{C} \longrightarrow \mathscr{C}$ is a natural transformation satisfying the coherence axioms

$$\begin{tikzcd}
a \arrow[r, "\nabla_a"] \arrow[d, "\nabla_a"'] & a\oplus a \arrow[d, "\iota_a\oplus\nabla_a"] \\
a\oplus a \arrow[r, "\nabla_a\oplus\iota_a"'] & a\oplus a\oplus a
\end{tikzcd}
\qquad
\begin{tikzcd}
a \arrow[r, "\nabla_a"] \arrow[dr, "\nabla_a"'] & a\oplus a \arrow[d, "\rho_{a,a}"] \\
& a\oplus a
\end{tikzcd}
\qquad
\begin{tikzcd}
a\oplus b \arrow[r, "\nabla_a\oplus\nabla_b"] \arrow[dr, "\nabla_{a\oplus b}"'] & a\oplus a\oplus b\oplus b \arrow[d, "\iota_a\oplus\rho_{a,b}\oplus\iota_b"] \\
& a\oplus b\oplus a\oplus b
\end{tikzcd}$$

[1] Given functors $F, G : \mathscr{A} \rightarrow \mathscr{B}$, a *transformation* $\tau : F \Rightarrow G : \mathscr{A} \rightarrow \mathscr{B}$ is a family $\tau = \{\tau_a : F(a) \rightarrow G(a) \mid a \in O_{\mathscr{A}}\}$ of arrows in $\mathscr{B}$ indexed by objects of $\mathscr{A}$. We say that τ is *natural* if $\tau_a; G(f) = F(f); \tau_b$ for every arrow $f : a \rightarrow b$ in $\mathscr{A}$ and an *isomorphism* if all its components τ_a's are so.

While symmetric monoidal categories are a staple of theoretical computer science, at least since the seminal work by Meseguer and Montanari [26], g-monoidal were introduced for the functorial semantics of partial algebras [9]. In the present context, they turns out to capture the idempotency of our additive operator.

Making each object s a co-monoid object (more precisely, a co-semigroup object, since the unity e and the arrows $a \to e$ are missing [23]) and requiring the naturality of ∇ provide some form of idempotency for the sum of terms and it is connected to the irredundancy requirement for concrete processes. Still, we have to ban the identity $t = t \oplus t$: that would be problematic as it would lead to undesirable equalities between terms, and on this we will offer further remarks in the concluding section.

Definition 23 (*GPROPs*) A GPROP is a 5-tuple $\langle S, \mathscr{C}, \_\oplus\_, \gamma, \nabla\rangle$, where $\langle \mathscr{C}, \_\oplus\_, \gamma, \nabla\rangle$ is a g-monoidal category such that the objects of $\mathscr{C}$ are elements of S^* and $\oplus$ acts as string concatenation on objects, i.e., $s_1 \oplus s_2 = s_1 s_2$.

Bimonoidal categories, and their coherence laws, have been considered early on in the literature [22]. Recently they surfaced, sometimes with the name *rig* or *semiring* categories, in the definition of models for quantum programming [17].

We introduce dioidal categories in order to obtain a categorical counterpart of dioids, i.e., semirings where the additive operator is idempotent. In the following, we consider dioidal categories satisfying an additional requirement.

Definition 24 (*Bipermutative and dioidal categories*) A *bipermutative* category is a 6-tuple $\langle \mathscr{C}, \_\oplus\_, \rho, \_\otimes\_, \gamma, \psi\rangle$, where $\langle \mathscr{C}, \_\oplus\_, \rho\rangle$ and $\langle \mathscr{C}, \_\otimes\_, \gamma\rangle$ are permutative categories such that the objects $(a \oplus b) \otimes c$ and $(a \otimes c) \oplus (b \otimes c)$ coincide and $\psi : \__1 \otimes (\__2 \oplus \__3) \Rightarrow (\__1 \otimes \__2) \oplus (\__1 \otimes \__3) : \mathscr{C} \times \mathscr{C} \times \mathscr{C} \longrightarrow \mathscr{C}$ is a natural isomorphism satisfying the coherence axioms

$$\begin{tikzcd}
(a \oplus b) \otimes c \arrow[r, "="] \arrow[d, "(t_1 \oplus t_2) \otimes t_3"'] & (a \otimes c) \oplus (b \otimes c) \arrow[d, "(t_1 \otimes t_3) \oplus (t_2 \otimes t_3)"] \\
(a_1 \oplus b_1) \otimes c_1 \arrow[r, "="'] & (a_1 \otimes c_1) \oplus (b_1 \otimes c_1)
\end{tikzcd}$$

$$\begin{tikzcd}
a \otimes (b \oplus c) \arrow[r, "\psi_{a,b,c}"] \arrow[d, "\gamma_{a,b\oplus c}"'] & (a \otimes b) \oplus (a \otimes c) \arrow[d, "\gamma_{a,b} \oplus \gamma_{a,c}"] \\
(a_1 \oplus b_1) \otimes c_1 \arrow[r, "="'] & (a_1 \otimes c_1) \oplus (b_1 \otimes c_1)
\end{tikzcd}
\qquad
\begin{tikzcd}
(a \oplus b) \otimes c \arrow[r, "="] \arrow[d, "\rho_{a,b} \otimes c"'] & (a \otimes c) \oplus (b \otimes c) \arrow[d, "\rho_{a \otimes c, b \otimes c}"] \\
(b \oplus a) \otimes c \arrow[r, "="'] & (b \otimes c) \oplus (a \otimes c)
\end{tikzcd}$$

$$\begin{tikzcd}
(a \oplus b) \otimes (c \oplus d) \arrow[r, "\psi_{a \oplus b,c,d}"] \arrow[d, "="'] & ((a \oplus b) \otimes c) \oplus ((a \oplus b) \otimes d) \arrow[r, "="] & (a \otimes c) \oplus (b \otimes c) \oplus (a \otimes d) \oplus (b \otimes d) \arrow[d, "(\iota_a \otimes \iota_c) \oplus \rho_{b \otimes c, a \otimes d} \oplus (\iota_b \otimes \iota_d)"] \\
(a \otimes (c \oplus d)) \oplus (b \otimes (c \oplus d)) \arrow[rr, "\psi_{a,c,d} \oplus \psi_{b,c,d}"'] & & (a \otimes c) \oplus (a \otimes d) \oplus (b \otimes c) \oplus (b \otimes d)
\end{tikzcd}$$

Finally, a *dioidal category* is a 7-tuple $\langle \mathscr{C}, \_\oplus\_, \rho, \nabla, \_\otimes\_, \gamma, \psi\rangle$, where $\langle \mathscr{C}, \_\oplus\_, \rho, \nabla\rangle$ is a g-monoidal category and $\langle \mathscr{C}, \_\oplus\_, \rho, \_\otimes\_, \gamma, \psi\rangle$ is a bipermutative category satisfying the coherence axiom

$$\begin{tikzcd}
a \otimes b \arrow[r, "\nabla_a \otimes \iota_b"] \arrow[dr, "\nabla_{a \otimes b}"'] & (a \oplus a) \otimes b \arrow[d, "="] \\
& (a \otimes b) \oplus (a \otimes b)
\end{tikzcd}$$

$$\frac{a \in S^*}{\iota_a : a \to a \in \mathbf{AP}(R)} \qquad \frac{t \in T}{t : \zeta_0(t) \to \zeta_1(t) \in \mathbf{AP}(R)} \qquad \frac{a, b \in S^*}{\gamma_{a,b} : ab \to ba \in \mathbf{AP}(R)}$$

$$\frac{\chi_1 : a \to c, \chi_2 : c \to b \in \mathbf{AP}(R)}{\chi_1 ; \chi_2 : a \to b \in \mathbf{AP}(R)} \qquad \frac{\chi_1 : a_1 \to b_1, \chi_2 : a_2 \to b_2 \in \mathbf{AP}(R)}{\chi_1 \otimes \chi_2 : a_1 a_2 \to b_1 b_2 \in \mathbf{AP}(R)}$$

Fig. 8 The set of rules generating $\mathbf{AP}(R)$

$$\chi; \iota_b = \chi = \iota_a; \chi \qquad (\chi_1 \otimes \chi_2); (\chi_3 \otimes \chi_4) = (\chi_1; \chi_3) \otimes (\chi_2; \chi_4) \qquad \iota_{ab} = \iota_a \otimes \iota_b$$

$$\gamma_{a,b}; \gamma_{b,a} = \iota_{ab} \qquad \gamma_{a,bc} = (\gamma_{a,b} \otimes \iota_c); (\iota_b \otimes \gamma_{a,c}); \qquad \gamma_{a_1,a_2}; (\chi_2 \otimes \chi_1) = (\chi_1 \otimes \chi_2); \gamma_{b_1,b_2}$$

Fig. 9 The set of axioms quotienting $\mathbf{AP}(R)$

Once again, we simplified the definition dropping the requirement of a unity object with respect to $\oplus$. Observe that the underlying semiring of objects is only right-distributive, as witnessed by the top axioms above for bipermutative categories. Requiring also left-distributivity would boil down to have that $\psi_{a,b,c}$ is an identity, and this seems unreasonable unless the $\oplus$ operator is commutative (see [25, Section 12]).

We next introduce the PROP counterpart of bipermutative and dioidal categories.

Definition 25 (*BI- and DPROPS*) A *BIPROP* is a 8-tuple $\langle S, \mathscr{C}, \_ \oplus \_, \rho, \_ \otimes \_, \gamma, \psi \rangle$, where $\langle S, \mathscr{C} \_ \oplus \_, \rho, \_ \otimes \_, \gamma, \psi \rangle$ is a bipermutative category such that the objects of $\mathscr{C}$ are elements of $S^{\langle * \rangle}$, i.e., non-empty lists $\langle s_1, \ldots, s_n \rangle$ of elements $s_i \in S^*$, $\oplus$ acts as list concatenation on objects, and $\otimes$ acts as a right distributive string concatenation on objects, i.e., $\langle s_1, \ldots, s_m \rangle \otimes \langle t_1, \ldots, t_n \rangle = \langle s_1 t_1, \ldots, s_1 t_n, \ldots, s_m t_1, \ldots, s_m t_n \rangle$.

A *DPROP* is a BIPROP based on a dioidal category.

4.2 Categories of processes

This section introduces two categories, out of the transitions of a pre-net, which are shown to form a PROP and a GPROP, respectively: they capture the monoidal structure of deterministic and non-deterministic processes of pre-nets.

We start by recalling the categorical characterisation of deterministic processes for pre-nets. This is just a rephrasing in the PROP-terminology of the results developed in [7] along the lines of the original proposal in [11] for Petri nets.

Lemma 2 *Let R be a pre-net. Then, the 4-tuple $\langle S, \mathbf{P}(R), \_ \otimes \_, \gamma \rangle$ is a PROP.*

In other terms, the category of concatenable processes has a monoidal structure, and actually, a permutative one, where the objects are freely generated. In fact, the arrows are freely generated, too. The formal statement is reported below.

Proposition 4 (The PROP of deterministic processes) *Let R be a pre-net and $\mathbf{AP}(R)$ the permutative category whose objects are elements of S^*, $\otimes$ acts as string concatenation, and arrows are generated according to the rules in Fig. 8, subject to the axioms in Fig. 9. Then, $\langle S, \mathbf{AP}(R), \_ \otimes \_, \gamma \rangle$ is isomorphic (as a PROP) to $\langle S, \mathbf{P}(R), \_ \otimes \_, \gamma \rangle$.*

Since the composition operator is partial, some of the axioms in Fig. 9 are required to hold only whenever both sides are defined. The objects of $\mathbf{AP}(R)$ are thus non-empty strings representing sources and targets of deterministic processes. Its arrows are equivalence classes of concrete elements generated by the set of inference rules in Fig. 8, modulo the equations making it a permutative category.

$$\frac{a \in S^{\langle * \rangle}}{\iota_a : a \to a \in \mathbf{ANP}(R)} \qquad \frac{\eta_1 : a \to c, \eta_2 : c \to b \in \mathbf{ANP}(R)}{\eta_1 ; \eta_2 : a \to b \in \mathbf{ANP}(R)}$$

$$\frac{a, b \in S^{\langle * \rangle}}{\rho_{a,b} : a \oplus b \to b \oplus a \in \mathbf{ANP}(R)} \qquad \frac{a \in S^{\langle * \rangle}}{\nabla_a : a \to a \oplus a \in \mathbf{ANP}(R)}$$

$$\frac{\chi : a \to b \in \mathbf{AP}(R)}{\langle \chi \rangle : \langle a \rangle \to \langle b \rangle \in \mathbf{ANP}(R)} \qquad \frac{\eta_1 : a_1 \to b_1, \eta_2 : a_2 \to b_2 \in \mathbf{ANP}(R)}{\eta_1 \oplus \eta_2 : a_1 \oplus a_2 \to b_1 \oplus b_2 \in \mathbf{ANP}(R)}$$

Fig. 10 The set of rules generating $\mathbf{ANP}(R)$

$$\langle \chi_1 \rangle ; \langle \chi_2 \rangle = \langle \chi_1 ; \chi_2 \rangle \qquad \iota_{\langle a \rangle} = \langle \iota_a \rangle$$

$$\eta ; \iota_b = \eta = \iota_a ; \eta \qquad (\eta_1 \oplus \eta_2) ; (\eta_3 \oplus \eta_4) = (\eta_1 ; \eta_3) \oplus (\eta_2 ; \eta_4) \qquad \iota_{a \oplus b} = \iota_a \oplus \iota_b$$

$$\rho_{a,b} ; \rho_{b,a} = \iota_{a \oplus b} \qquad \rho_{a, b \oplus c} = (\rho_{a,b} \oplus \iota_c) ; (\iota_b \oplus \rho_{a,c}) \qquad \rho_{a_1, a_2} ; (\eta_2 \oplus \eta_1) = (\eta_1 \oplus \eta_2) ; \rho_{b_1, b_2}$$

$$\nabla_a ; \rho_{a,a} = \nabla_a \quad \nabla_a ; (\iota_a \oplus \nabla_a) = \nabla_a ; (\nabla_a \oplus \iota_a) \quad \nabla_{a \oplus b} = (\nabla_a \oplus \nabla_b) ; (\iota_a \oplus \rho_{a,b} \oplus \iota_b) \quad \nabla_a ; (\eta \oplus \eta) = \eta ; \nabla_{b \oplus b}$$

Fig. 11 The set of axioms quotienting $\mathbf{ANP}(R)$

Being isomorphic as a PROP means that the isomorphism functor preserves the symmetries γ and the elements of S. Moreover, the monoidal operator on objects is preserved on-the-nose, as expected since in both categories the objects are strings of places. In fact, the functor is freely induced by the function mapping the arrow t to the corresponding concatenable process δ_t for each transition t of the pre-net.

As already mentioned, the result for deterministic processes is well-known in the Petri net literature. Looking at the recent categorical literature, it might be derived by the concrete characterisation of the free PROP built out of an hyper-graph, as detailed in [19].[2] A direct proof can be carried out by obtaining a normal forms for the arrows of $\mathbf{AP}(R)$, and then providing a concrete function between those arrows and processes of $\mathbf{CP}(R)$. Indeed, we are going to proceed this way for non-deterministic processes.

Firstly, we observe that the category of non-deterministic processes is a GPROP.

Lemma 3 *Let R be a pre-net. Then, the 5-tuple $\langle S^*, \mathbf{NP}(R), \_ \oplus \_, \nabla, \rho \rangle$ is a GPROP.*

Proof The proof just consists in a lengthy but straightforward check of the naturality of ∇ and ρ, and of the validity of the coherence laws. □

Thus, also the category of concatenable non-deterministic processes of a pre-net has a monoidal structure, and in fact, a g-monoidal one, where the objects are freely generated: the constants are the strings of S, and the objects are lists of strings. Once again, the arrows are freely generated, too. The formal statement is reported below: its proof will be carried out throughout the rest of this subsection.

Theorem 1 (The GPROP of non-deterministic processes) *Let R be a pre-net and $\mathbf{ANP}(R)$ the g-monoidal category whose objects are elements of $S^{\langle * \rangle}$, $\oplus$ acts as list concatenation on objects, and arrows are generated according to the rules in Fig. 10, subject to the axioms in Fig. 11. Then, $\langle S^*, \mathbf{ANP}(R), \_ \oplus \_, \nabla, \rho \rangle$ is isomorphic (as a GPROP) to $\langle S^*, \mathbf{NP}(R), \_ \oplus \_, \nabla, \rho \rangle$.*

[2] Even if, despite the common usage, the terminology adopted in [19] is that of *mega*-graphs.

Axioms $\langle\chi_1\rangle; \langle\chi_2\rangle = \langle\chi_1; \chi_2\rangle$ and $\iota_{\langle a\rangle} = \langle\iota_a\rangle$ guarantee that the categorical structure of $\mathbf{AP}(R)$ is lifted to $\mathbf{ANP}(R)$.[3] Furthermore, axiom $\iota_{a\oplus b} = \iota_a \oplus \iota_b$ ensures that the identities in $\mathbf{ANP}(R)$ are obtained as composition of those in $\mathbf{AP}(R)$.[4]

As in the case of $\mathbf{AP}(R)$, being isomorphic as a GPROP means that the isomorphism functor preserves the symmetries, the duplicators, and the elements of S^*, and as before the monoidal operator on objects is preserved on-the-nose. In fact, the functor is built on top (i.e., it lifts) the functor mapping $\mathbf{AP}(R)$ into $\mathbf{P}(R)$, so that in fact it preserves the elements of S as well as the deterministic processes.

To the best of our knowledge, the result is new in the Petri net literature: no algebraic structure has been proposed for non-deterministic processes either for nets or for pre-nets. The proof is carried out below and it follows the same lines as the one for deterministic processes: it first obtains a normal form for the arrows of $\mathbf{ANP}(R)$, and then it provides a concrete function between such arrows and the processes of $\mathbf{NP}(R)$. A key point in the procedure is the collapsing phase that is required in the composition of non-deterministic processes, which is mimicked by the naturality of ∇, i.e., by the axiom $\nabla; (\eta\oplus\eta) = \eta; \nabla$. In order to get some intuition, let us consider the arrow $\eta = \langle t\rangle$ for some transition t. If we interpret arrows as concrete processes, the concatenation of ∇ followed by $\eta \oplus \eta$, as in the left-hand side of the naturality axiom, produces a redundant process consisting of a non-deterministic choice between two occurrences of transition t. Its collapsing is the irredundant process consisting of a single t whose target is duplicated and offered twice. Instead, the concatenation of η followed by ∇, as in the right-hand side of the axiom, produces immediately the latter irredundant process. We will comment further the issue of redundancy in Sect. 5.

We next proceed with the proof of Theorem 1. As a first step, along the lines of the decomposition for non-deterministic processes in Lemma 1, we can decompose arrows of $\mathbf{ANP}(R)$ in a "maximally redundant" shape. Before, we need to introduce a shorthand: given $a \in S^{\langle *\rangle}$, we let ∇^n_a denote the n-times duplication of a, uniquely obtained by induction as $\nabla^1_a = \iota_a$ and $\nabla^{k+1}_a = \nabla_a; (\nabla^k_a \oplus \iota_a)$.

Lemma 4 (Normal form) *Let R be a pre-net and $\eta : \langle s_1, \ldots, s_n\rangle \to \langle t_1, \ldots, t_m\rangle$ an arrow of* $\mathbf{ANP}(R)$.

1. *If $n = 1$, then η can be decomposed as*

$$\nabla^m_{\langle s_1\rangle}; (\langle\chi_1\rangle \oplus \cdots \oplus \langle\chi_m\rangle)$$

   such that $\chi_k \in \mathbf{AP}(R)$ for $k \in \{1, \ldots, m\}$.
2. *If $n > 1$ then η can be decomposed as*

$$(\eta_1 \oplus \cdots \oplus \eta_n); \rho$$

   with $\eta_i : \langle s_i\rangle \to \langle t_{i_1}, \ldots, t_{i_j}\rangle$, where $t_{i_1} \ldots t_{i_j}$ are strings occurring in $t_1 \ldots t_m$ such that $i_1 \leq i_2 \leq \cdots \leq i_j$.
   Moreover, ρ does not alter the relative order of the targets of the η_i, i.e., ρ is stable with respect to the equivalence $\Sigma^{h-1}_{j=0} i_j + k_1 \sim \Sigma^{h-1}_{j=0} i_j + k_2$, for $h \in [1, n]$ and $k_1, k_2 \in [1, i_h]$.

Proof We first focus on point (1). Observe that the arrows of $\mathbf{ANP}(R)$ are equivalence classes of terms built out of the rules in Fig. 10, quotiented by the axioms in Fig. 11. Objects of $\mathbf{ANP}(R)$ are tuples of strings. Since the generators (i.e., the arrows from $\mathbf{AP}(R)$) have a

3 In abstract data types terms, an order-sorted algebra with type $\mathbf{AP}(R)$ included in type $\mathbf{ANP}(R)$.

4 Indeed, also for ρ and ∇ would suffice to consider only those instances associated to the objects of $\mathbf{AP}(R)$, and consider the axioms involving them as definitions of derived operators.

unary source and target (i.e., both source and target are tuples consisting of a single string), it is immediate to see that all the symmetries can be moved to the far left. The desired shape is then obtained by observing that any arrow that originates from $a \in S^*$ and contains no generator from $\mathbf{AP}(R)$ is equated by the axioms to an arrow of the shape ∇_a^k.

Concerning point (2), the desired shape is readily recovered by exploiting the naturality of ∇ and ρ and the functoriality of $\oplus$. The fact that the number of components of η coincides with the number of strings in the originating object is due again to the fact that the arrows generating $\mathbf{ANP}(R)$ have a unary source.

Recalling, again, that arrows with unary source and containing no generators from $\mathbf{AP}(R)$ are equivalent arrows of the shape ∇_a^k, it follows that the requirement on the stability of the symmetry ρ can be satisfied. □

The normal form for the arrows of $\mathbf{ANP}(R)$ does mimic the one for non-deterministic processes in Lemma 1. Hence, as mentioned before, to a certain extent it describes the maximally non-deterministic shape: Each resource is duplicated as many times as needed, and then each copy is used for a single deterministic process (see also Sect. 5). E.g., a term $t; \nabla$ leads to the normal form $\nabla; (t \oplus t)$. This can be used to provide a proof for Theorem 1.

Proof of Theorem 1 We prove Theorem 1 by relying on Lemmas 4 and 1.

First, by Proposition 4, we can consider a functor $\mathscr{D} : \mathbf{AP}(R) \to \mathbf{P}(R)$, establishing an isomorphism between $\mathbf{AP}(R)$ and the category of deterministic processes $\mathbf{P}(R)$, seen as PROPs. Observing that the arrows of $\mathbf{ANP}(R)$ are built inductively, according to the rules of Fig. 10, using the arrows of $\mathbf{AP}(R)$ as generators, we deduce that such functor can be extended to a functor $\mathscr{N} : \mathbf{ANP}(R) \to \mathbf{NP}(R)$ preserving the GPROP structure, since the axioms of Fig. 11 are clearly preserved.

Now, a simple, but fundamental observation is that the functor $\mathscr{N}$ maps terms of $\mathbf{ANP}(R)$ in normal form (as in Lemma 4) to normal form decompositions of concatenable non-deterministic processes (as in Lemma 1). This is immediate since duplicators and permutations are mapped to the corresponding duplicator and permutation processess, and for each term of shape $\langle\chi\rangle$, with χ arrow in $\mathbf{P}(R)$, by construction $\mathscr{N}(\langle\chi\rangle) = \langle\mathscr{D}(\chi)\rangle$ with $\mathscr{D}(\chi)$ a concatenable deterministic process in $\mathbf{P}(R)$.

The mapping is clearly surjective, whence the functor $\mathscr{N}$ is full. Moreover if two normal form terms t_1 and t_2 in $\mathbf{ANP}(R)$ are mapped to the same concrete concatenable non-deterministic process in $\mathbf{NP}(R)$, necessarily they have the same "shape", i.e., for $i \in \{1, 2\}$, we have $t_i = (\eta_1^i \oplus \cdots \oplus \eta_n^i); \rho^i$, with $\eta_j^i = \nabla_{\langle s_j\rangle}^{m_j}; (\langle\chi_1^{i,j}\rangle \oplus \cdots \oplus \langle\chi_{m_j}^{i,j}\rangle)$. Since ρ^1 and ρ^2 are mapped to the same permutation process, they are equivalent $\rho^1 \sim \rho^2$. Additionally, exploiting the faithfulness of the functor from $\mathbf{AP}(R)$ into $\mathbf{P}(R)$ we conclude that for $j \in \{1, \ldots, n\}$, $h \in \{1, \ldots, m_j\}$ it holds $\chi_h^{1,j} = \chi_h^{2,j}$. Therefore $t_1 \sim t_2$. This allows us to conclude that the functor $\mathscr{N}$ is also faithful. (A detailed proof could be given as an adaptation of the proof of correspondence between term graphs and gs-monoidal categories in [8, Section 4]: see also Sect. 5).

Recalling that the categories $\mathbf{ANP}(R)$ and $\mathbf{NP}(R)$ have the same objects, and the functor is the identity on objects, we conclude that $\mathscr{N}$ is an isomorphism. □

4.3 A dioidal category for non-deterministic processes

In this section we show that the category of concatenable non-deterministic processes $\mathbf{NP}(R)$ is a dioidal one. We first prove that the free diodal category generated from the net, seen as a GPROP, is isomorphic to $\mathbf{ANP}(R)$, the free g-monoidal category built over concaten-

$$\frac{a \in S^{\langle * \rangle}}{\iota_a : a \to a \in \mathbf{ADP}(R)} \qquad \frac{\eta_1 : a \to c, \eta_2 : c \to b \in \mathbf{ADP}(R)}{\eta_1 ; \eta_2 : a \to b \in \mathbf{ADP}(R)}$$

$$\frac{a, b \in S^{\langle * \rangle}}{\rho_{a,b} : a \oplus b \to b \oplus a \in \mathbf{ADP}(R)} \qquad \frac{a \in S^{\langle * \rangle}}{\nabla_a : a \to a \oplus a \in \mathbf{ADP}(R)}$$

$$\frac{t \in T}{t : \langle \zeta_0(t) \rangle \to \langle \zeta_1(t) \rangle \in \mathbf{ADP}(R)} \qquad \frac{a, b \in S^{\langle * \rangle}}{\gamma_{a,b} : a \otimes b \to b \otimes a \in \mathbf{ADP}(R)}$$

$$\frac{\eta_1 : a_1 \to b_1, \eta_2 : a_2 \to b_2 \in \mathbf{ADP}(R)}{\eta_1 \otimes \eta_2 : a_1 \otimes a_2 \to b_1 \otimes b_2 \in \mathbf{ADP}(R)} \qquad \frac{\eta_1 : a_1 \to b_1, \eta_2 : a_2 \to b_2 \in \mathbf{ADP}(R)}{\eta_1 \oplus \eta_2 : a_1 \oplus a_2 \to b_1 \oplus b_2 \in \mathbf{ADP}(R)}$$

Fig. 12 The set of rules generating $\mathbf{ADP}(R)$

$$\psi_{\langle a \rangle, b, c} = \iota_{(\langle a \rangle \otimes b)} \oplus \iota_{(\langle a \rangle \otimes c)} \qquad \psi_{\langle a \rangle \oplus b, c, d} = (\psi_{\langle a \rangle, c, d} \oplus \psi_{b,c,d}); (\iota_{\langle a \rangle \otimes c} \oplus \rho_{\langle a \rangle \otimes d, b \otimes c} \oplus \iota_{b \otimes d})$$

Fig. 13 The natural transformation ψ

$$\eta ; \iota_b = \eta = \iota_a ; \eta \qquad (\eta_1 \oplus \eta_2);(\eta_3 \oplus \eta_4) = (\eta_1;\eta_3) \oplus (\eta_2;\eta_4) \qquad \iota_{a \oplus b} = \iota_a \oplus \iota_b \qquad \iota_{a \otimes b} = \iota_a \otimes \iota_b$$

$$\rho_{a,b};\rho_{b,a} = \iota_{a \oplus b} \qquad \rho_{a, b \oplus c} = (\rho_{a,b} \oplus \iota_c);(\iota_b \oplus \rho_{a,c}) \qquad \rho_{a_1,a_2};(\eta_2 \oplus \eta_1) = (\eta_1 \oplus \eta_2);\rho_{b_1,b_2}$$

$$\nabla_a;\rho_{a,a} = \nabla_a \quad \nabla_a;(\iota_a \oplus \nabla_a) = \nabla_a;(\nabla_a \oplus \iota_a) \quad \nabla_{a \oplus a} = (\nabla_a \oplus \nabla_a);(\iota_a \oplus \rho_{a,a} \oplus \iota_a) \quad \nabla_a;(\eta \oplus \eta) = \eta;\nabla_{b \oplus b}$$

$$\gamma_{a,b};\gamma_{b,a} = \iota_{a \otimes b} \qquad \gamma_{a, b \otimes c} = (\gamma_{a,b} \otimes \iota_c);(\iota_b \otimes \gamma_{a,c}) \qquad \gamma_{a_1,a_2};(\eta_2 \otimes \eta_1) = (\eta_1 \otimes \eta_2);\gamma_{b_1,b_2}$$

$$\gamma_{a, b \oplus c} = \psi_{a,b,c};(\gamma_{a,b} \oplus \gamma_{a,c}) \qquad (\eta_1 \otimes \eta_3) \oplus (\eta_2 \otimes \eta_3) = (\eta_1 \oplus \eta_2) \otimes \eta_3$$

$$\rho_{a \otimes b, a \otimes c} = \rho_{a,b} \otimes \iota_c \qquad \nabla_{a \otimes b} = \nabla_a \otimes \iota_b$$

Fig. 14 The set of axioms quotienting $\mathbf{ADP}(R)$

able deterministic processes. Then we conclude exploiting the algebraic characterisation of $\mathbf{NP}(R)$, which in Theorem 1 is shown to be isomorphic to $\mathbf{ANP}(R)$. In terms of concrete processes, the result provides a notion of parallel composition for non-deterministic processes, extending the parallel composition for deterministic processes. The formal statement is reported below, its proof will be carried out throughout the rest of this subsection.

Proposition 5 (The DPROP of non-deterministic processes) *Let R be a pre-net and $\mathbf{ADP}(R)$ the dioidal category whose objects are elements of $S^{\langle * \rangle}$, where $\oplus$ acts as list concatenation on objects, $\otimes$ acts as a right distributive string concatenation on objects, i.e., $\langle a_1, \ldots, a_m \rangle \otimes \langle b_1, \ldots, b_n \rangle = \langle a_1 b_1, \ldots, a_1 b_n, \ldots, a_n b_1 \ldots, a_n b_m \rangle$, and the arrows are generated according to the rules in Fig. 12, subject to the axioms in Fig. 14. Then, $\langle S^*, \mathbf{ADP}(R), \_ \oplus \_, \nabla, \rho \rangle$ is isomorphic (as a GPROP) to $\langle S^*, \mathbf{NP}(R), \_ \oplus \_, \nabla, \rho \rangle$.*

Note that the natural transformation ψ in Fig. 13, which makes $\mathbf{ADP}(R)$ a dioidal category, is actually a derived operator: it is a chosen permutation with source $a \otimes (b \oplus c)$ and target $(a \otimes b) \oplus (a \otimes c)$. It allows us to simplify the presentation of the axioms concerning γ in Fig. 14.

Lemma 5 ($\mathbf{ADP}(R)$ is dioidal) *Let R be a pre-net. Then, $\langle S, \mathbf{ADP}(R), \_ \oplus \_, \nabla, \rho, \_ \otimes \_, \gamma, \psi \rangle$ is a DPROP.*

Proof We just need to show that ψ is natural and that $\psi_{a \oplus b, c, d}$ verifies the decomposition property stated as a coherence axiom in Definition 24. The latter can be shown by induction on the length of $a \oplus b$, while the former is proved by the diagram below.

$$
\begin{tikzcd}
a_1\otimes(a_2\oplus a_3) \arrow[r, "\gamma_{a_1,a_2\oplus a_3}"] \arrow[d, "\psi_{a_1,a_2,a_3}"'] \arrow[rrr, bend left, "\eta_1\otimes(\eta_2\oplus\eta_3)"] & (a_2\oplus a_3)\otimes a_1 \arrow[r, "(\eta_2\oplus\eta_3)\otimes\eta_1"] \arrow[d, "="] & (b_2\oplus b_3)\otimes b_1 \arrow[r, "\gamma_{b_2\oplus b_3,b_1}"] \arrow[d, "="'] & b_1\otimes(b_2\oplus b_3) \arrow[d, "\psi_{b_1,b_2,b_3}"] \\
(a_1\otimes a_2)\oplus(a_1\otimes a_3) \arrow[r, "\gamma_{a_1,a_2}\oplus\gamma_{a_1,a_3}"'] \arrow[rrr, bend right, "(\eta_1\otimes\eta_2)\oplus(\eta_1\otimes\eta_3)"'] & (a_2\otimes a_1)\oplus(a_3\otimes a_1) \arrow[r, "(\eta_2\otimes\eta_1)\oplus(\eta_3\otimes\eta_1)"'] & (b_2\otimes b_1)\oplus(b_3\otimes b_1) \arrow[r, "\gamma_{b_2,b_1}\oplus\gamma_{b_3,b_1}"'] & (b_1\otimes b_2)\oplus(b_1\otimes b_3)
\end{tikzcd}
$$

□

Proposition 5 states that the category of concrete non-deterministic processes has a dioidal structure, induced by the g-monoidal one. The proof relies on the way the arrows of $\mathbf{ADP}(R)$ are generated, and it is a consequence of the lemma below.

Lemma 6 (Reducing the dioidal complexity) *Let R be a pre-net. Then, $\langle S^*, \mathbf{ADP}(R), \_\oplus\_, \nabla, \rho\rangle$ is isomorphic (as a GPROP) to $\langle S^*, \mathbf{ANP}(R), \_\oplus\_, \nabla, \rho\rangle$.*

Proof We first note that there exists a functor $\mathbf{ANP}(R) \to \mathbf{ADP}(R)$ that preserves the GPROP structure. In fact, each term obtained via the rules for $\mathbf{ANP}(R)$ in Fig. 10, which in turn is built on top of those for $\mathbf{AP}(R)$ in Fig. 8, is indeed generated also by the rules for $\mathbf{ADP}(R)$ in Fig. 12. The same goes for the corresponding sets of axioms, thus the functor does exist.

The next step is to prove that such a functor is full and faithful.

As for fullness, it suffices to note that each term generated in $\mathbf{ADP}(R)$ can be expressed via the axioms of Fig. 14 as a composition of terms in $\mathbf{ANP}(R)$. This is true for the generators: it just needs to be shown for $\gamma_{a,b}$, which is now defined for all lists of strings $a, b \in S^{\langle * \rangle}$. To this aim observe that the second argument of γ can be simplified by repeatedly using the right-most axiom of the line before the last in Fig. 14, namely $\gamma_{a,b\oplus c} = \psi_{a,b,c}; (\gamma_{a,b} \oplus \gamma_{a,c})$. Similarly, the first argument of γ can be simplified by the equality $\gamma_{a\oplus b,c}; \psi_{c,a,b} = \gamma_{c,a} \oplus \gamma_{c,b}$, which is derived by rewriting the same axiom. Combining the two, we prove that $\gamma_{a\oplus b,c\oplus d}$ identifies a chosen permutation with source $(a \oplus b) \otimes (c \oplus d)$ and target $(a \otimes b) \oplus (c \otimes d)$ in $\mathbf{ANP}(R)$.

We then move to show that the decomposition holds for those terms whose top operator is $\otimes$. First of all, distributivity. Since $\otimes$ is strictly right-distributive on arrows, as it is for objects, it suffices to show that left-distributivity holds, up-to isomorphism. This boils down to prove the naturality of ψ, which is shown in the proof of Lemma 5.

The two axioms of the last line in Fig. 14 guarantee the decomposition of $\rho_{a,b} \otimes \iota_c$ and $\nabla_a \otimes \iota_b$. By the naturality of γ we have $\gamma_{a\oplus b,c}; (\iota_c \otimes \rho_{a,b}) = (\rho_{a,b} \otimes \iota_c); \gamma_{b\oplus a,c}$, and finally we have $(\iota_b \otimes \nabla_a); \psi_{b,a,a} = \nabla_{b\otimes a}$, as shown by the diagram below.

$$
\begin{tikzcd}
 & b\otimes(a\oplus a) \arrow[dr, "\gamma_{b,a\otimes a}"] \arrow[rr, "\iota_b\otimes(\iota_a\oplus\iota_a)"] & & \\
b\otimes a \arrow[ur, "\iota_b\otimes\nabla_a"] \arrow[r, "\gamma_{b,a}"] \arrow[dr, "\iota_b\otimes\iota_a"'] & a\otimes b \arrow[r, "\nabla_a\otimes\iota_b"] \arrow[d, "\gamma_{a,b}"] \arrow[dr, "\nabla_{a\otimes b}"'] & (a\oplus a)\otimes b \arrow[r, "\gamma_{a\oplus a,b}"] \arrow[d, "="] & b\otimes(a\oplus a) \arrow[d, "\psi_{b,a,a}"] \\
 & b\otimes a \arrow[rr, bend right, "\nabla_{b\otimes a}"'] & (a\otimes b)\oplus(a\otimes b) \arrow[r, "\gamma_{a,b}\oplus\gamma_{a,b}"'] & (b\otimes a)\oplus(b\otimes a)
\end{tikzcd}
$$

As for faithfulness, note that the extra axioms involving γ, ρ, and ∇ actually boil down to a coherence property stating the one-to-one correspondence between the arrows in $\mathbf{ADP}(\emptyset)$ with source a and target b, and the arrows in $\mathbf{ANP}(\emptyset)$ with same source and target (which correspond to surjective functions up-to permutation from b to a, as it occurs to the ρ and ∇ fragment: see e.g. [8, Lemma 15]). The only relevant axiom is thus right-distributivity. However, it can be considered as a right-to-left rewriting rule, resulting in a normal form for any equivalence class of terms. □

We finally go back to the main result of this subsection: it is now straightforward.

Proof of Proposition 5 Immediate consequence of Lemma 6 and Theorem 1. □

5 On neutral elements and redundancy

In this section we elaborate on two themes. First we discuss the choice of omitting the neutral element for the monoidal operators representing parallel and non-deterministic composition. We argue that while for parallel composition the presence of the neutral element would be inessential, in the case of non-deterministic composition it has interesting consequences and a natural interpretation in terms of concrete processes. Secondly, we elaborate on the choice of requiring irredundancy on processes, outlining the theory that would arise omitting this requirement and discussing the corresponding graphical interpretation.

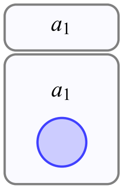
5.1 Neutral elements

Adding the neutral element (corresponding to the empty marking) for parallel composition would not require relevant changes in the algebraic theory of deterministic processes. At the level of pre-nets, this change would be naturally imply the relaxation of the constraint of non-emptyness for the pre- and post-sets of transitions. This would not hinder the notion of process but the corresponding theory of (concrete) concatenable processes would be more complex. First of all, the presence of transitions with empty pre-set invalidates the current notion of canonical process. In fact, the canonical name for a transition is made unique by using the list of places in the pre-set of the transition. It would be possible to adapt the notion of canonical process with a special treatment of minimal transitions with empty pre-set, but the resulting construction would be notably more complex. This additional complication would not be suitably rewarded since, from an operational point of view, allowing for empty pre-sets just implies the unrealistic possibility of firing an unbounded number of transition in parallel. Allowing empty post-sets would instead be less problematic. The main difference would be that cuts lose their operational interpretation as snapshots of a computation, since the firing of transitions with empty post-set would not be witnessed by maximal places. However, since the addition of a sink place for any transition with empty post-set results in a operationally equivalent net, it is quite standard to require that also post-sets are not empty.

For the above reasons we do not deepen further the theory that would arise by adding the neutral element for parallel composition and, in the rest of this section, we continue to work without such neutral element and with the assumption that transitions have non-empty pre- and post-sets.

The insertion of the neutral element for non-deterministic composition is instead more interesting. It fits naturally with a relaxed notion of non-deterministic process, where the result of some deterministic sub-computations can be discarded.

Fig. 15 The process $!_a$



Definition 26 (*Loose concatenable non-deterministic process*) Let R be a pre-net. A *loose concatenable non-deterministic process* η for R is a triple of lists $\langle \boldsymbol{\alpha}, \boldsymbol{\pi}, \boldsymbol{\omega} \rangle$ with

- $\boldsymbol{\pi} = \pi_1 \cdots \pi_n$ is a list of (pairwise disjoint) non-deterministic processes;
- $\boldsymbol{\alpha} = \alpha_1 \ldots \alpha_n$ is a list of strings such that $\alpha_i \in {S_{O_i}}^*$ and $\mu(\alpha_i) = \min(\pi_i)$;
- $\boldsymbol{\omega} = \omega_1 \ldots \omega_\ell$ is a list of strings such that $\omega_j \in {S_{O_i}}^*$ for some $i \in \{1, \ldots, n\}$ and $\mu(\omega_j) \in cuts(\pi_i)$.

The source of η is the list $\zeta_0(\eta) = {\pi_1}^*(\alpha_1) \ldots {\pi_n}^*(\alpha_n)$, i.e., the list of the sources of the component processes, while the target of η is $\zeta_1(\eta) = u_1 \ldots u_\ell$, where $u_j = {\pi_i}^*(\omega_j)$ if $\mu(\omega_j) \in cuts(\pi_i)$.

With respect to Definition 14, the lists $\boldsymbol{\alpha}$, $\boldsymbol{\pi}$, and $\boldsymbol{\omega}$ are possibly empty and the requirement that W is a covering is removed. Consider e.g. the non-deterministic process of Fig. 5. Should the target be $\langle \{e_1, f_1, d_1, b_2\}\{g_2\}\{f_2, g_1\} \rangle$, it would not be a covering anymore, since the post-set $\{b_4\}$ of the transition z does not appear in the target. With such an additional freedom we have two new relevant basic processes (which play a role in the correspondence with the algebraic characterisation), namely the process $!_a$ that discards its source and has an empty list of targets, depicted in Fig. 15, and the empty process, consisting of an empty list of components, with empty lists of sources and targets. The corresponding obviously defined category of loose concatenable non-deterministic processes is denoted as $\mathbf{NP}_l(R)$.

We now move to the algebraic characterisation of loose processes. As a start, note that with respect to Definition 20, the monoidal operator $\oplus$ for non-deterministic composition is equipped with a neutral object ϵ such that $\epsilon \oplus a = a$ and the coherence axiom $\iota_\epsilon \oplus t = t$ is verified. Correspondingly, a permutative category is a monoidal one verifying the coherence axiom $\rho_{\epsilon,\epsilon} = \iota_\epsilon$ (from which e.g. $\rho_{\epsilon,a} = \iota_a$ follows).

The notion of bipermutative category as given in Definition 24 also generalises smoothly: assuming $\otimes$ still being a semi-group category and ϵ the neutral element of $\oplus$, it suffices to require that the objects $\epsilon \otimes a$ and ϵ coincide and impose the coherence axioms $\iota_\epsilon \otimes t = \iota_\epsilon$ (from which e.g. $\gamma_{\epsilon,a} = \iota_\epsilon$ follows).

The adaptation of the notion of g-monoidal and hence of dioidal categories is slightly more complex, but, interestingly enough, the corresponding notions are well-studied in the literature (see e.g. [8,9], where they are used for modelling term graphs and providing a functorial semantics for multi-algebras). We provide an explicit definition below, basically taken from [9]. It includes also some variations of g-monoidal categories which are going to be at hand in the following subsection.

Definition 27 (*Diodal categories*) A *gs-monoidal category* is a 6-tuple $\langle \mathscr{C}, \_\oplus\_, \epsilon, \rho, \nabla, ! \rangle$, where $\langle \mathscr{C}, \_ \oplus \_, \epsilon, \rho \rangle$ is a permutative category and $! : \_{}_1 \Rightarrow \epsilon : \mathscr{C} \longrightarrow \mathscr{C}$ and $\nabla : \_{}_1 \Rightarrow$

$\_{}_1 \oplus \_{}_1 : \mathscr{C} \longrightarrow \mathscr{C}$ are transformations satisfying the coherence axioms $!_\epsilon = \iota_\epsilon$ and

$$\begin{tikzcd} a \arrow[r, "\nabla_a"] \arrow[d, "\iota_a"'] & a \oplus a \arrow[d, "\iota_\epsilon \oplus !_a"] \\ a \arrow[r, "="'] & \epsilon \oplus a \end{tikzcd} \qquad \begin{tikzcd} a \oplus b \arrow[r, "!_a \oplus !_b"] \arrow[dr, "!_{a \oplus b}"'] & \epsilon \oplus \epsilon \arrow[d, "="] \\ & \epsilon \end{tikzcd}$$

$$\begin{tikzcd} a \arrow[r, "\nabla_a"] \arrow[d, "\nabla_a"'] & a \oplus a \arrow[d, "\iota_a \oplus \nabla_a"] \\ a \oplus a \arrow[r, "\nabla_a \oplus \iota_a"'] & a \oplus a \oplus a \end{tikzcd} \qquad \begin{tikzcd} a \arrow[r, "\nabla_a"] \arrow[dr, "\nabla_a"'] & a \oplus a \arrow[d, "\rho_{a,a}"] \\ & a \oplus a \end{tikzcd} \qquad \begin{tikzcd} a \oplus b \arrow[r, "\nabla_a \oplus \nabla_b"] \arrow[dr, "\nabla_{a \oplus b}"'] & a \oplus a \oplus b \oplus b \arrow[d, "\iota_a \oplus \rho_{a,b} \oplus \iota_b"] \\ & a \oplus b \oplus a \oplus b \end{tikzcd}$$

A gs-monoidal category is *g-monoidal* if ∇ is a natural transformation.

A *pre-dioidal (dioidal) category* is a 9-tuple $\langle \mathscr{C}, \_ \oplus \_, \epsilon, \rho, \nabla, !, \_ \otimes \_, e, \gamma, \psi \rangle$, where $\langle \mathscr{C}, \_ \oplus \_, \epsilon, \rho, \nabla, ! \rangle$ is a gs-monoidal (g-monoidal) category and $\langle \mathscr{C}, \_ \oplus \_, \epsilon, \rho, \_ \otimes \_, \gamma, \psi \rangle$ is a bipermutative category satisfying the coherence axioms

$$\begin{tikzcd} a \otimes b \arrow[r, "\nabla_a \otimes \iota_b"] \arrow[dr, "\nabla_{a \otimes b}"'] & (a \oplus a) \otimes b \arrow[d, "="] \\ & (a \otimes b) \oplus (a \otimes b) \end{tikzcd} \qquad \begin{tikzcd} a \otimes b \arrow[r, "!_a \otimes \iota_b"] \arrow[dr, "!_{a \otimes b}"'] & \epsilon \otimes b \arrow[d, "="] \\ & \epsilon \end{tikzcd}$$

We then extend the notion of GPROP such that its objects are now possibly empty lists of non-empty strings over S. The correspondence with loose processes can still be recovered, as it is stated by the lemma below, which generalises Lemma 3.

Lemma 7 *Let R be a pre-net. Then, the 7-tuple $\langle S^*, \mathbf{NP}_l(R), \_ \oplus \_, \epsilon, \rho, \nabla, ! \rangle$ is a GPROP.*

We leave to the reader the reformulation of Theorem 1 and Proposition 5.

5.2 Redundancy

We hinted in several places in the paper to the fact that most of the notions concerning the theory of pre-net processes could be actually recast in terms of suitable graphs *with interfaces*, hence in the end in terms of PROPs. In this section we comment further on such correspondence and observe that it can become even tighter by relaxing the requirement of irredundancy on pre-net processes.

A deterministic process $\pi : O \to R$ can be naturally seen as a *typed* graph, i.e., as a graph morphism from a directed acyclic hyper-graph, the occurrence pre-net O, to R, which plays the role of a type graph. The requirement of absence of forward and backward conflicts in O amounts to imposing that in the correspoding graph any node has at most one outgoing or one ingoing, respectively, edge. Concatenable processes have, additionally, an interface, i.e., in graph theoretical terms, a pair of morphisms $In \to O$, $Out \to O$ for discrete graphs In and Out. In a well-defined process, In and Out are just the minimal and the maximal places, respectively, of the process.

The axioms on the category $\mathbf{AP}(R)$ of concatenable processes (see Proposition 4) have an immediate graphical interpretation: they are aimed at establishing a one-to-one correspondence between the graphical and the set-theoretical presentation of processes. Hence Proposition 4 is just rephrasing in terms of PROPs the classical results about deterministic processes [10,26], i.e., that deterministic processes are the arrows of the free symmetric

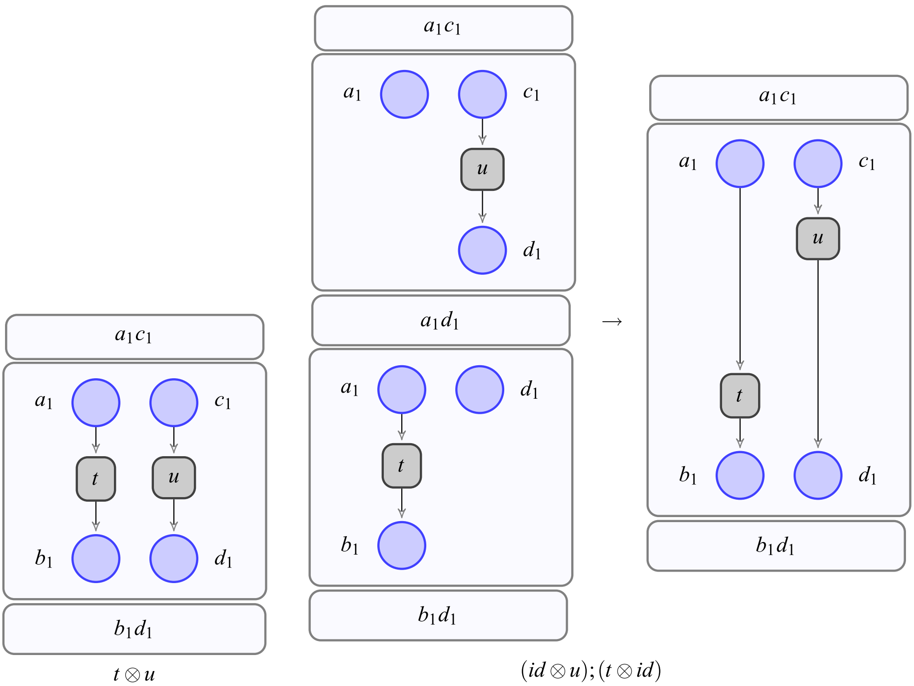


Fig. 16 Direct interpretation of the interchange axiom

(strict) monoidal category generated by R (and see also the recent survey on graphical languages of monoidal categories [33]). In other words, the axioms are immaterial since they are hard-wired in the visual representation. Consider, e.g., two transitions $t : a \to b$ and $u : c \to d$ and the derived law $\delta_t \otimes \delta_u = (id \otimes \delta_u); (\delta_t \otimes id)$, as shown by Fig. 16.

Following the seminal work on traced monoidal categories [20], there has been a renewed interest in graphical languages for concurrent computational systems, as witnessed by the survey mentioned above. However, while the characterisation of parallel composition as a monoidal operation is well-understood, the situation is not equally established for those formalisms mixing parallel and non-deterministic behaviour. Indeed, tight connections has been established between such mixed structures and e.g. suitable notions of multi-relations [34], but, to the best of our knowledge there has been so far no proposal for a purely graphical interpretation of bipermutative categories. Thus, we believe that our correspondence result for pre-net processes highlights the role of such categories in the graphical representation of non-determinism.

With hindsight, our proposal is in the end quite simple: a non-deterministic process is just a list of graphs with interface $\alpha_i \to O_i, \omega_j \to O_i$. We can then focus on a single non-deterministic process, and thus most axioms have an immediate graphical interpretation, in the sense they identify terms having the same graphical representation as processes. The only exception is the naturality of ∇, i.e., $t; \nabla = \nabla; (t \oplus t)$: as already mentioned, it is related to the irredundancy conditions for non-deterministic processes and aims at identifying redundant processes with the same collapsing.

This can be made more precise by elaborating a bit on an algebraic characterisation of possibly irredundant processes. Let $\mathbf{RNP}_l(R)$ be the category of concatenable redundant non-

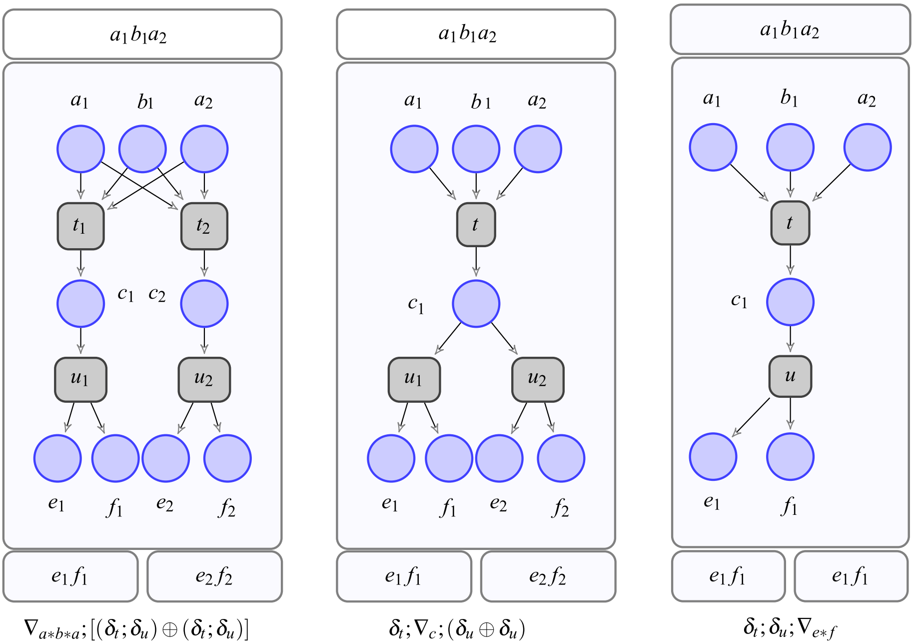


Fig. 17 Three redundant processes

deterministic processes: parallel composition is the same as for non-deterministic processes, while sequential composition is as in Definition 16, but avoiding the collapsing phase. Then we have the following.

Lemma 8 *Let R be a pre-net. Then, the 7-tuple $\langle S^*, \mathbf{RNP}_l(R), \_ \oplus \_, \epsilon, \rho, \nabla, !\rangle$ is a GSPROP.*

Relying on the representation result concerning gs-monoidal categories [8] we now could reformulate Theorem 1 and give an algebraic characterisation of the redundant non-deterministic processes of a net R as the arrows of the free gs-monoidal category (see Definition 27) built over a net R (as well as an equivalent of Proposition 5 for pre-dioidal categories).

As an example, consider the three (possibly redundant) processes for the pre-net R_0 in Fig. 1 depicted in Fig. 17 (the left-most and central ones are redundant, while the right-most, which has been presented already in Fig. 7c, is irredundant).

The three terms ∇_{a*b*a}; $[(\delta_t;\, \delta_u)\oplus(\delta_t;\, \delta_u)]$, δ_t; ∇_c; $(\delta_u \oplus \delta_u)$, and δ_t; δ_u; ∇_{e*f} correspond to processes that just differ for the degree of sharing. They would be collapsed to the same irredundant process (the right-most) once the irredundancy condition is imposed and indeed the three terms are equated by the axioms defining $\mathbf{ANP}(R)$.

In other terms, while possibly redundant processes can be seen as graphs and the categorical axioms naturally corresponds to the graphical interpretation, the irredundancy condition and the related collapsing phase corresponds to maximising the sharing, thus identifying redundant processes differing only for the degree of sharing. The move from redundant to irredundant processes is formalised in terms of a functor from the GSPROP category of redundant processes in Lemma 8 to the GPROP category of irredundant processes in Lemma 7.

Such a functor exists as the latter category differs from the former just for the addition of an axiom, i.e., the naturality of ∇.

6 Conclusions and further works

Along the lines of the seminal paper [26], our work offers an algebraic presentation for the non-deterministic computations of Petri (pre-)nets.

As a first step we introduced concatenable non-deterministic processes for Petri nets, building on the (non-concatenable) processes originally defined by Engelfriet [12]. Then we investigated the algebraic structure of the resulting category, showing that it has a tight link with a class of bimonoidal categories, which we called dioidal categories: the category of non-deterministic processes of a net arises as the dioidal category freely generated from the net itself. To the best of our knowledge, putting dioidal categories into the limelight represents a small addition to the categorical lore. The idea of mixing two monoidal structures for representing parallel and non-deterministic behaviours in (possibly distributed) systems has surfaced in the literature. In particular, our formalism shares with semiringal categories [18] the emphasis on the distributivity laws between the two monoidal operators. Among the former proposals for the characterisation of non-deterministic net computations, the most reminiscent of our solution is the one based on linear categories [24], i.e., categories with a monoidal and a cartesian structure. It is precisely our choice of working with the less structured bimonoidal categories that allows us to establish the central result of our paper: a functorial bijection between the concatenable non-deterministic processes of a net and the arrows of a suitable free category built out of it.

An established trend in formal methods has been the adoption of graphical presentations for modelling concurrent and distributed systems, such as network algebras [35] and bigraphs [27]. With similar aims, a larger attention has been recently devoted to visual languages for (extensions of) monoidal categories [33]. The main tool for such monoidal case are PROPs, whose connection with linear algebra and flow/signal graphs has been the focus of much work (see e.g. [5]). Our introduction of GPROPs can thus be considered as a further validation of the paradigm, offering a visual tool for bipermutative/dioidal categories, and pave the way to further research on the area. Indeed, considering also the characterisation of redundant processes in Sect. 5.2, we believe that redundancy can be characterised by giving a direction $\nabla; (t \oplus t) \to t; \nabla$ to the naturality axioms and then introducing a suitable graph rewriting system, along the lines of what has been done for Frobenius categories [4].

As for further refinements on the categorical model, as e.g. the self-dual category for modelling processes of contextual nets proposed in [14], let us just mention that we toyed with the idea of capturing the idempotency of $\oplus$ by making ∇ a natural isomorphism (hence, more in tune with the algebraic notion of dioids). The concrete description of concatenable non-deterministic processes does not allow it, since there would be no possible interpretation for the arrow $(\nabla_a)^{-1} : a \oplus a \to a$. However, this is not unfortunate, since the naturality of ∇ would make the diagram below commute

$$\begin{tikzcd}
a\oplus a \arrow[r, "(\nabla_a)^{-1}"] \arrow[d, "\nabla_a\oplus\nabla_a"'] & a \arrow[r, "\nabla_a"] & a\oplus a \\
a\oplus a\oplus a\oplus a \arrow[rr, "\iota_a\oplus\gamma_{a,a}\oplus\iota_a"'] & & a\oplus a\oplus a\oplus a \arrow[u, "(\nabla_a)^{-1}\oplus(\nabla_a)^{-1}"']
\end{tikzcd}$$

We would infer that $(t_1 \oplus t_2); (t_3 \oplus t_4)$ is equated by functoriality to $(t_1; t_3) \oplus (t_2; t_4)$ and by naturality to $(t_1; t_3) \oplus (t_1; t_4) \oplus (t_2; t_3) \oplus (t_2; t_4)$, while those terms should intuitively represent different non-deterministic processes. Idempotency and functoriality look like clashing properties for the $\oplus$ operator, and we could not let the latter go.

As a final remark, we believe that, from the point of view of net theory, the categorical presentation of non-deterministic processes can also contribute to a clearer presentation of the relation between the (deterministic) process semantics and the unfolding semantics. In fact, given a net R and an initial state u, consider the slice category $u \downarrow \mathbf{NP}(R)$. Relying also on the results in [12], it should be easy to show that such category is a pre-order, whose ideal completion is a lattice having as top element the unfolding of R (from u) as defined in [2].

Acknowledgements We are indebted to Professor Peter May for the interaction and the fruitful discussions on bimonoidal categories, as well as to the reviewers for their remarks and pointers to the literature.

References

1. Baldan, P., Bonchi, F., Gadducci, F., Monreale, G.V.: Modular encoding of synchronous and asynchronous interactions using open Petri nets. Sci. Comput. Program. **109**, 96–124 (2015)
2. Baldan, P., Bruni, R., Montanari, U.: Pre-nets, read arcs and unfolding: a functorial presentation. In: Wirsing, M., Pattinson, D., Hennicker, R. (eds.) Algebraic Development Techniques (WADT 2002), Lecture Notes in Computer Science, vol. 2755, pp. 145–164. Springer (2003)
3. Baldan, P., Gadducci, F.: Petri nets are dioids. In: Meseguer, J., Rosu, G. (eds.) Algebraic Methodologies and Software Technology (AMAST 2008), Lecture Notes in Computer Science, vol. 5140, pp. 51–66. Springer (2008)
4. Bonchi, F., Gadducci, F., Kissinger, A., Sobocinski, P., Zanasi, F.: Rewriting modulo symmetric monoidal structure. In: Grohe, M., Koskinen, E., Shankar, N. (eds.) Logic in Computer Science (LICS 2016), pp. 710–719. ACM, New York (2016)
5. Bonchi, F., Sobocinski, P., Zanasi, F.: Full abstraction for signal flow graphs. In: Rajamani, S.K., Walker, D. (eds.) Principles of Programming Languages (POPL 2015), pp. 515–526. ACM, New York (2015)
6. Bruni, R., Gadducci, F., Montanari, U.: Normal forms for algebras of connections. Theor. Comput. Sci. **286**(2), 247–292 (2002)
7. Bruni, R., Meseguer, J., Montanari, U., Sassone, V.: Functorial models for Petri nets. Inf. Comput. **170**(2), 207–236 (2001)
8. Corradini, A., Gadducci, F.: An algebraic presentation of term graphs, via gs-monoidal categories. Appl. Categ. Struct. **7**(4), 299–331 (1999)
9. Corradini, A., Gadducci, F.: A functorial semantics for multi-algebras and partial algebras, with applications to syntax. Theor. Comput. Sci. **286**, 293–322 (2002)
10. Degano, P., Meseguer, J., Montanari, U.: Axiomatizing net computations and processes. In: Logic in Computer Science (LICS 1989), pp. 175–185. IEEE Computer Society (1989)
11. Degano, P., Meseguer, J., Montanari, U.: Axiomatizing the algebra of net computations and processes. Acta Inform. **33**(7), 641–667 (1996)
12. Engelfriet, J.: Branching processes of Petri nets. Acta Inform. **28**(6), 575–591 (1991)
13. Esparza, J., Heljanko, K.: Unfoldings: a partial order approach to model checking. Springer, Berlin (2008)
14. Gadducci, F., Montanari, U.: Axioms for contextual net processes. In: Larsen, K., Skyum, S., Winskel, G. (eds.) Automata, Languages and Programming (ICALP 1998), Lecture Notes in Computer Science, vol. 1443, pp. 296–308. Springer (1998)
15. Goltz, U., Reisig, W.: The non-sequential behaviour of Petri nets. Inf. Control **57**(2/3), 125–147 (1983)
16. Gorrieri, R.: Process algebras for Petri nets: the alphabetization of distributed systems. Springer, Berlin (2017)
17. Green, A., Altenkirch, T.: From reversible to irreversible computations. In: Selinger, P. (ed.) Quantum Programming Languages (QPL 2006), Electronic Notes in Theoretical Computer Science, vol. 210, pp. 65–74. Elsevier, Amsterdam (2008)
18. Grosu, R., Lucanu, D., Stefanescu, G.: Mixed relations as enriched semiringal categories. Univers. Comput. Sci. **6**(1), 112–129 (2000)
19. Hackney, P., Robertson, M.: On the category of PROPs. Appl. Categ. Struct. **23**(4), 543–573 (2015)

20. Joyal, A., Street, R., Verity, D.: Traced monoidal categories. Math. Proc. Camb. Philos. Soc. **119**(3), 447–468 (1996)
21. Lack, S.: Composing PROPs. Theory Appl. Categ. **13**(9), 147–163 (2004)
22. Laplaza, M.: Coherence for distributivity. In: Coherence in Categories, Lecture Notes in Mathematics, vol. 281, pp. 29–72. Springer (1972)
23. Mac Lane, S.: Categories for the Working Mathematician. Springer, Berlin (1971)
24. Martì-Oliet, N., Meseguer, J.: From Petri nets to linear logic through categories: a survey. Found. Comput. Sci. **2**(4), 297–399 (1991)
25. May, J.: The construction of E_∞ ring spaces from bipermutative categories. Geom. Topol. Monogr. **16**, 283–330 (2009)
26. Meseguer, J., Montanari, U.: Petri nets are monoids. Inf. Comput. **88**(2), 105–155 (1990)
27. Milner, R.: The Space and Motion of Communicating Agents. Cambridge University Press, Cambridge (2009)
28. Nielsen, M., Plotkin, G., Winskel, G.: Petri nets, event structures and domains, part 1. Theor. Comput. Sci. **13**, 85–108 (1981)
29. Petri, C.: Kommunikation mit automaten. Ph.D. Thesis, Institut für Instrumentelle Matematik, Bonn (1962)
30. Reisig, W.: Petri nets: an introduction. Springer, Berlin (1985)
31. Sangiorgi, D., Walker, D.: The Pi-Calculus: A Theory of Mobile Processes. Cambridge University Press, Cambridge (2001)
32. Sassone, V.: An axiomatization of the category of Petri net computations. Math. Struct. Comput. Sci. **8**(2), 117–151 (1998)
33. Selinger, P.: A survey of graphical languages for monoidal categories. Spring. Lect. Notes Phys. **13**(813), 289–355 (2011)
34. Stefanescu, G.: Reaction and control I. Mixing additive and multiplicative network algebras. Log. J. IGPL **6**(2), 348–369 (1998)
35. Stefanescu, G.: Network Algebra. Springer, Berlin (2000)
36. Winskel, G.: Event structures. In: Brauer, W., Reisig, W., Rozenberg, G. (eds.) Petri Nets: Applications and Relationships to Other Models of Concurrency, Lecture Notes in Computer Science, vol. 255, pp. 325–392. Springer (1987)