Two algebraic process semantics for contextual nets

Roberto Bruni[1] and Vladimiro Sassone[2]

[1] Dipartimento di Informatica, Università di Pisa, Italia.
[2] Dipartimento di Matematica e Informatica, Università di Catania, Italia.
bruni@di.unipi.it, vs@dmi.unict.it

Abstract We show that the so-called 'Petri nets are monoids' approach initiated by Meseguer and Montanari can be extended from ordinary place/transition Petri nets to *contextual* nets by considering suitable non-free monoids of places. The algebraic characterizations of net concurrent computations we provide cover both the *collective* and the *individual* token philosophy, *uniformly* along the two interpretations, and coincide with the classical proposals for place/transition Petri nets in the absence of read-arcs.

Introduction

The basic features common to any 'flavour' of Petri net [28] essentially are that states are (multi)sets of distributed, abstract resources, and that actions only involve the coordination of local parts of the state, as they can consume some of the resources available and release fresh resources. Accordingly, a computation can be described abstractly as a partial order of events in which any two events are either causally dependent – when one could not have been executed without a resource provided by the other – or concurrent – when they could have happened in any order, because they affect independent subsystems. These features make net models suitable for representing in a satisfactory way *concurrent* and *distributed systems* in many interdisciplinary applications.

Meseguer and Montanari in [22,23] (and successively in [12,13,31,32,6,14] several authors) have recasted these facts in algebraic terms to unveil properties of net computations and, especially, of the intrinsic concurrency of the net model. The underlying idea of the so-called 'Petri nets are monoids' approach is to lift the algebraic structure of states to the level of computations, so that the distribution of the resources is reflected on the performed actions, analogously to what happens in *rewriting logic* [20,21], in *structured transition systems* [11] and in *tile logic* [17,4]. In the case of ordinary place/transition Petri nets (PT nets), states are multisets of places, or equivalently, elements of the free commutative monoid over the set of places. Moreover, a computation can be obviously composed with any computation that originates from the same state in which the first ends, yielding a computation that is the concatenation

Research supported by CNR Integrated Project *Metodi e Strumenti per la Progettazione e la Verifica di Sistemi Eterogenei Connessi mediante Reti di Comunicazione*; by Esprit Working Groups *CONFER2* and *COORDINA*; and by MURST project *TOSCa: Tipi, Ordine Superiore e Concorrenza*.

of the two. Hence, computations possess by nature an intrinsic (partial) operation of 'sequential' composition that gives rise to a *category* – arrows are computations, identities representing unused tokens. Lifting the monoidal structure of states to the category of computations results in a *monoidal category of computations*, where the functoriality law of the monoidal tensor product expresses a basic fact about the *true concurrency* of the model. Namely, that in any computation the relative order in which two concurrent actions are executed is always immaterial. In fact, if α_1 and α_2 are computations such that α_i, for $i = 1, 2$, originates in u_i and leads to v_i (written $\alpha_i : u_i \to v_i$), then

$$(\alpha_1 \oplus id_{u_2}); (id_{v_1} \oplus \alpha_2) = \alpha_1 \oplus \alpha_2 = (id_{u_1} \oplus \alpha_2); (\alpha_1 \oplus id_{v_2}),$$

where $\oplus$ is the tensor product (modeling concurrent composition of computations) originated from multiset union on states, $\_;\_$ is the operation of sequential composition, and the id_{u_i}, id_{v_i} are idle components of computations, with e.g., $id_{u_1}; \alpha_1 = \alpha_1 = \alpha_1; id_{v_1}$.

The extensive use of PT nets has given rise to different schools of thought concerning their semantic interpretation. In particular, the main distinction is drawn between *collective* and *individual token philosophies* (see e.g. [33]). According to the collective token philosophy (*CTph*), one is not interested in distinguishing among different *tokens* in the same place (i.e., among instances of the same resource), because all such tokens are operationally equivalent. However, tokens may have different origins and histories, carrying different *causality* information and hence consuming one instance rather than another, can make the difference from being causally dependent or not on some previous event. The point of view of the individual token philosophy (*ITph*) is that these causal dependencies may well form an essential information that should not be discarded when, e.g., flow analysis is concerned. Of course, causal dependencies may influence the degree of concurrency in abstract computations, and therefore *CTph* and *ITph* lead to quite different concurrent semantics.

For ordinary PT nets the algebraic approach has been pursued under both philosophies, characterizing different kinds of *net processes*, ranging from Best and Devillers commutative processes [3] (that support the *CTph*) to concatenable processes [13,31] and strongly concatenable processes [32] (that support the *ITph*). Note that the *ITph* relies on a tensor product which can be commutative only *up to* a monoidal natural isomorphism. Therefore, the algebraic approach requires some special mechanism in order to accommodate the lifting of the (commutative) monoidal structure of states. It is worth mentioning that the algebraic approach under the *ITph* is completely straightforward for the recent proposal of *pre-nets* [6] whose states are based on strings rather than multisets. From this point of view, the approach initiated by Meseguer and Montanari is completely general and can be applied to more general net models where, e.g., tokens are some kind of more complex data [25,15].

Several extensions of the basic PT net paradigm have been considered in the literature that either increase the expressive power or give a better representation of existing phenomena. This paper focuses on extending the 'Petri nets are monoids' approach to *contextual nets*, also known as nets with *read-arcs*, or *condition-arcs*, or *test-arcs* [10,27,19,35]. The motivating idea behind 'read-arcs' is that of *reading* resources *without consuming* them, thus providing a way of modeling multiple concurrent accesses to the same resource. Using ordinary PT nets such readings must be rendered

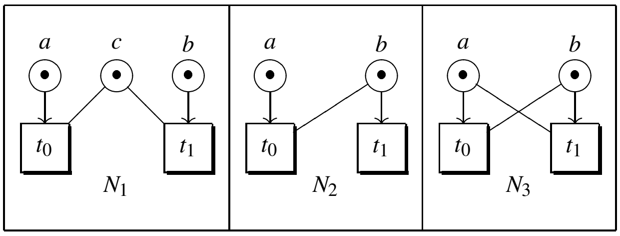

Figure 1.

as *self-loops*, and this imposes an unwanted sequentialization of concurrent readings. On the contrary, with contextual nets, besides pre and post-sets, transitions also have '*contexts*', that is resources that are necessary for the enabling but are *not* affected by the firing. Contextual nets have found applications e.g., to transaction serializability in databases [30], concurrent constraint programming [26], and asynchronous systems [34].

Independently of *CTph* and *ITph*, for contextual nets several different approaches have been proposed that differ for the way in which contexts are read. For example, let us consider the nets N_1, N_2 and N_3 in Figure 1, taken from [35]. (As usual, places are represented by circles, tokens by black bullets, transitions by boxes, pre- and post-sets by directed weighted arcs, and contexts by undirected weighted arcs, with unary weights always omitted.) According to the semantic interpretation of [27], the transitions t_0 and t_1 can fire concurrently in N_1, but neither in N_2 nor in N_3, since the basic assumption is that a token cannot be read and consumed in the same step. In [19], instead, the concurrent step is allowed for all three nets, the basic assumption being that t_0 and t_1 can both start together, read the context tokens, and need them *not* while the actions take place. Besides its possible merits, we find this interpretation not fully convincing as, for instance, in N_3 we would end up in a state that cannot be reached by any firing sequence. The basic assumption of [35] that firings have duration leads to consider ST-traces, where explicit *transition-starts* and *transition-ends* events are fired. Hence N_2 can start t_0 and then t_1 before t_0 completes, allowing the concurrent step $\{t_0, t_1\}$ (with the hypothesis that t_0 starts first). On the contrary, in N_3 if either t_0 or t_1 starts, then the context for the other transition is consumed and the concurrent step is forbidden. We follow the interpretation of [27] that fits better our understanding of contexts.

Contextual nets and collective token philosophy. The algebraic theory for PT nets developed under the *CTph* is well consolidated, and the relationships between its *computational*, *algebraic* and *logical* interpretations are by now very clear [5]. Starting with the classical 'token-game' semantics, many computational models for Petri nets have been proposed that follow the *CTph*. In particular, the *commutative processes* of Best and Devillers [3] reconcile the 'diamond' equivalence (cf. § 1.1) on firing and step sequences, and express very nicely the concurrency of the model. They also admit an exact algebraic representation by means of the universal construction $\mathcal{T}(\_)$ that yields *strictly symmetric strict monoidal categories* from the category of PT nets. More precisely, given a PT net N, the objects of $\mathcal{T}(N)$ are markings and its arrows correspond to the commutative processes of N [23,13].

Surprisingly, *CTph* semantics for contextual nets have received poor attention in the literature, not only for what concerns the algebraic treatment. Whether because the problem has been underestimated, or simply because the *ITph* is more fascinating, we cannot tell. In any case, we think that it is useful to address this discrepancy with the semantics of ordinary PT nets. Moreover, although one can easily extend the diamond equivalence to firing sequences on contextual nets, the formalization of a good *algebraic* model is not at all straightforward. Inspired by a suggestion made by Meseguer in [21], we give here a satisfactory treatment of this issue. The idea is to consider monoidal categories with a commutative tensor product taken – differently from the case of PT nets – over a *non-free* monoid of places. In particular, we regard each token a as an *atom* (for lack of a better analogy) that can emit 'negative' particles $\bar{a}$ (*electrons*) while keeping track of their number, i.e., as in [21], we assume that for all $k \in \mathbb{N}$, $a = a^k \oplus k \cdot \bar{a}$, where a^k represents an atom that has released exactly k particles to the environment.

Replacing context arcs on a with self-loop arcs on $\bar{a}$, we are able to give an axiomatic construction of a monoidal category whose arrows between standard markings (i.e., containing no negative particles) are (isomorphic to) the concurrent computations of the net according to the *CTph*. A key ingredient for this result to hold is the so-called *maximum sharing hypothesis*, an axiom expressing that concurrent readings can *always* be seen as sharing the same token, a fundamental idea in *CTph*.

Contextual nets and individual token philosophy. Building on the notion of *process* introduced by Goltz and Reisig in [18], several authors have shown that the semantics of nets in the *ITph* can still be understood in terms of *symmetric monoidal categories*, where the tensor product, this time denoted by $\_ \otimes \_$, is commutative only up to a monoidal natural isomorphism γ called *symmetry* (for strictly symmetric monoidal categories the transformation γ is just the identity). In particular, a simple variation of Goltz-Reisig processes called *concatenable processes* is introduced in [13] (see also [31]), which admits sequential composition and yields a symmetric monoidal category $\mathcal{P}(N)$ for each net N. Note that $\otimes$ is commutative on the objects of $\mathcal{P}(N)$. A refined version of concatenable processes is given by strongly concatenable processes [32] where origins and destinations are totally ordered (as opposed to the orderings of origins and destinations of concatenable processes that are indexed by the places). Also several unfolding semantics (see e.g. [36,24]) have been proposed that give a denotational interpretation of the interplay between concurrency, causality and nondeterminism.

For contextual nets both the process and the unfolding approaches have been studied [27,8,9,2,1], giving a satisfactory understanding of the computational model via the introduction of *asymmetric event structures*. The algebraic approach, however, has been pursued only in a recent paper by Gadducci and Montanari [16] using *match-share* categories. Their basic idea is that, together with symmetries, two additional auxiliary constructors must be present: one for *duplicating* tokens and one for *matching* them. Formally, for each place a the auxiliary arrows $\nabla_a : a \to a \otimes a$ and $\Delta_a : a \otimes a \to a$ are added to the computational model (and suitably axiomatized, by letting e.g.,

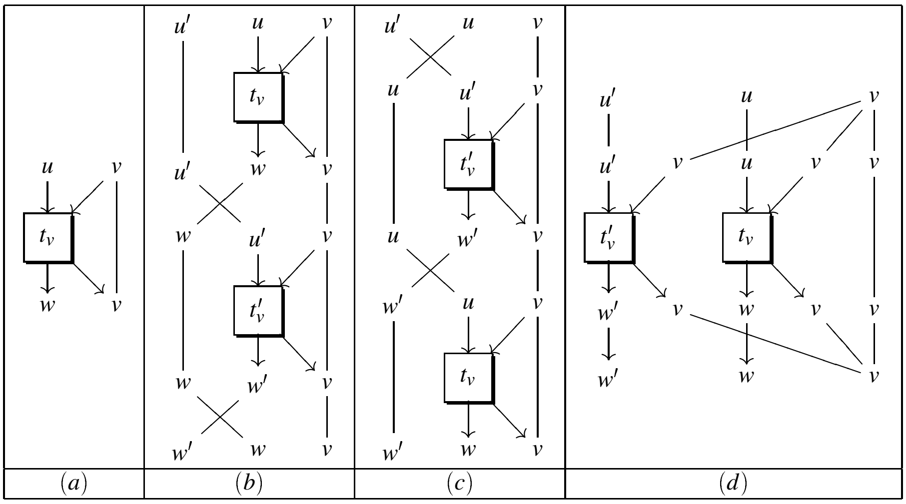

Figure 2.

$\nabla_a;\Delta_a = id_a$ and $\nabla_a;\gamma_{a,a} = \nabla_a$ with id_a the identity arrow on a and $\gamma_{a,a}$ the symmetry that swaps two tokens in a). Read-arcs can then be replaced by self-loops (i.e., if the transition t consumes u, reads v and produces w, then one considers a derived transition $t_v\colon u \otimes v \to w \otimes v$), and reading without consuming modeled by duplicating the context, firing the transition concurrently with an idle copy of the context, and then matching the idle copy with the corresponding produced tokens (i.e., by considering the arrow $\widehat{t_v} = (id_u \otimes \nabla_v);(t_v \otimes id_v);(id_w \otimes \Delta_v)$ illustrated in Figure 2(a)). Multiple concurrent access is achieved by producing via duplication – and then absorbing via matching – enough copies of the context. In [16], a suitable axiomatization of duplicators and matchers is introduced and proved to represent faithfully the basic fact about concurrent access: steps sharing the same context, but otherwise disjointly enabled, can execute concurrently or in any interleaved order with *no* noticeable difference (e.g., using the notation above, the term $(id_{u'} \otimes \widehat{t_v});(\gamma_{u',w} \otimes id_v);(id_w \otimes \widehat{t'_v});(\gamma_{w',w} \otimes id_v)$, illustrated in Figure 2(b), for t' that consumes u', reads v and produces w', is equivalent to $(\gamma_{u',u} \otimes id_v);(id_u \otimes \widehat{t'_v});(\gamma_{u,w'} \otimes id_u);(id_{w'} \otimes \widehat{t_v})$ in Figure 2(c), and both admit a normal form where the subterms t_v and t'_v are executed concurrently, as illustrated in Figure 2(d)).

The main drawback of this approach is that the initial model contains *too many* arrows and, therefore, in order to obtain a bijection with contextual processes one has to carve a suitable subcategory. Although the arrows of this subcategory can be characterized by inspecting their structure, the lack of a global correspondence somehow weakens the framework. We aim at improving the approach of [16] starting from the observation that the unwanted arrows are due to *redundant* information in the model. In fact, once a context token is read by a transition we know the ‘real’ token it is connected to: the one duplication was applied to. Hence, the match operation, needed for expressing concurrent readings, does not add any further information and may introduce inconsistent behaviors. For example, given two tokens in the place a, one can first duplicate both and then match each copy of the first token with a copy of the second token: it should be evident that the resulting arrow (written $(\nabla_a \otimes \nabla_a);(id_a \otimes \gamma_{a,a} \otimes id_a);(\Delta_a \otimes \Delta_a)$) is meaningless from the computational viewpoint, unless the two tokens represent the same context. We overcome this problem by extending to the *ITph* the approach proposed for the *CTph* in the first part of the paper.

The key of our proposal is to *regulate* the use of symmetries on the markings so to *forbid* the swapping of a a^k and an adjacent $a^{\bar{}}$. This prevents the migration of electrons from atom to atom, as it might happen in the *CTph* and in [16]. The absence of electron migration represents, in the *ITph*, a sort of dual to the maximum sharing hypothesis, that we call *exact sharing hypothesis*. Most notably, the restriction is imposed simply by omitting the corresponding symmetries from the model. And reintroducing them would in fact result in a redundant framework perfectly analogous to the one provided by match-share categories. Observe that this yields a monoidal category that, formally speaking, is *not* symmetric anymore: we allow only selected commutations by explicitly including selected symmetries. These will include, of course, all the symmetries between standard markings (i.e., those in which tokens have released no particles), and will exclude all those that may lead to confuse the causal histories of tokens. Our main result is that, again, the arrows between standard markings are in bijection with a slight refinement of contextual processes, called *concatenable contextual processes*. In this, it is crucial that the model be able to treat particles in different ways depending on the context. On the one hand, according to the *ITph*, we need to distinguish between $a^{\bar{}}$ released by different atoms, but on the other hand, similarly to the *CTph*, we want to identify those particles generated by the same a. This is the precise content of our exact sharing hypothesis, as formalized by a new axiom that we call (Δ).

Origin and structure of the paper. This paper builds on the work reported in [7]. Besides extending *loc. cit.* by detailed examples and proofs of the main results, we improve its treatment of the *ITph* in many respects. In particular, in [7] we relied on a distinction between forward and backward contexts, realized through a second kind of electron, $a_{\text{-}}$, in addition to $a^{\bar{}}$. Moreover, differently from here, our representation result was phrased in terms of *strongly* concatenable contextual processes. Axiom (Δ) is instrumental in these improvements, and is first introduced here.

In Section 1 we recall some basics about contextual nets and the algebraic semantics of PT nets. In Sections 2 and 3 we define algebraic semantics for contextual nets under both the *CTph* and the *ITph*, providing original characterization results for commutative and concatenable contextual processes. We remark that in the absence of read-arcs, our semantics coincide with the classical ones.

1 Preliminaries

1.1 Contextual nets

Contextual nets were introduced for extending PT nets with the 'read without consume' operation [10,27,19,35]. The states of contextual nets are called *markings* and represent distributions of resources (*tokens*) in typed repositories (*places*). Given the set of places S, markings can be seen as multisets $u: S \to \mathbb{N}$, where $u(a)$ denotes the number of tokens that place a carries in u. The set of finite multiset on S is the free commutative monoid on S. We denote it by $S^{\oplus}$, and indicate multiset inclusion, difference and union by $\subseteq$, $\oplus$ and $\ominus$, respectively, with $u \ominus v$ defined only for $v \subseteq u$. For k a natural number and u a multiset, $k \cdot u$ is the multiset such that $(k \cdot u)(a) = k \cdot u(a)$ for all a. We denote by $\lfloor u \rfloor$ the underlying set of u, that can be seen as the multiset such that $\lfloor u \rfloor(a) = 1$ if $u(a) > 0$ and $\lfloor u \rfloor(a) = 0$ otherwise.

Definition 1. *A* contextual net *N is a tuple $(S, T, \partial_0, \partial_1, \varsigma)$, where S is the set of places, T is the set of transitions, $\partial_0, \partial_1 : T \to S^{\oplus}$ are the pre and post-set functions, and $\varsigma: T \to S^{\oplus}$ is the context function. Besides the usual assumption that $\varsigma(t)$ and $\partial_0(t) \oplus \partial_1(t)$ are disjoint for each transition t, we assume that $\varsigma(t)$ is a set.*

Informally, $\partial_0(t) \oplus \varsigma(t)$ is the minimum amount of resources that t requires to be enabled. Of these resources, those in $\partial_0(t)$ are retrieved and consumed, while those in $\varsigma(t)$ are just read and left on their repositories. When t has accomplished its task, it returns $\partial_1(t)$ fresh tokens and releases the context. Only at this point other transitions will be able to consume the tokens in $\varsigma(t)$, whereas they can use the same context concurrently with t.

Definition 2. *Let u and v be markings, and X a finite multiset of transitions of a contextual net $N = (S, T, \partial_0, \partial_1, \varsigma)$. We say that u evolves to v under the* step *X, in symbols $u\,[X\rangle\, v$, if the transitions in X are concurrently enabled at u, i.e., $\lfloor \bigoplus_{t \in T} \varsigma(t) \rfloor \oplus \bigoplus_{t \in T} X(t) \cdot \partial_0(t) \subseteq u$, and*

$$v = u \ominus \left(\bigoplus_{t \in T} X(t) \cdot \partial_0(t) \right) \oplus \bigoplus_{t \in T} X(t) \cdot \partial_1(t).$$

A step sequence *from u_0 to u_n is a sequence $u_0\,[X_1\rangle\, u_1 \ldots u_{n-1}\,[X_n\rangle\, u_n$.*

Thus the execution of the step X requires that the marking u contains at least *all* the tokens in the preconditions $\partial_0(t)$ of transitions $t \in X$ plus at least *one* token for each place that is used as context by some transition in X. This matches the intuition that a token can be used as context by many transitions at the same time. From the point of view of concurrency, the fact that transitions in X are executed in a step means that they can be equivalently executed in any order. Thus, likewise ordinary PT nets, step sequences for contextual nets can be considered up to the equivalence induced by the *diamond transformation* relation $\_ \diamond \_$ defined by $u\,[X \oplus Y\rangle\, v \diamond u\,[X\rangle\, u_1\,[Y\rangle\, v$ for any step $u\,[X \oplus Y\rangle\, v$ (and suitable u_1). The *diamond equivalence* is the reflexive, symmetric, transitive and sequences concatenation closure of the relation $\_ \diamond \_$.

Definition 3. *Given a contextual net N, the strictly symmetric strict monoidal category (cf. § 1.2) of* contextual commutative processes *$\mathcal{CT}(N)$ has the markings of N as objects, its step sequences, taken modulo the diamond equivalence, as arrows, and composition is given by sequence concatenation.*

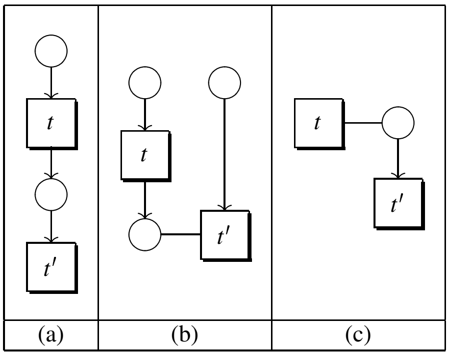

Figure 3. Three situations in which t (immediately) precedes t'.

In the *ITph*, computations are commonly described in terms of structures representing the causal relationships between event occurrences. In the case of nets, this is fruitfully formalized through the following notion of process. We remark that these notions are conservative extension of the corresponding notions for ordinary PT nets, to which they reduce in the absence of read-arcs. The relation $\_ \nearrow \_$ – used by Baldan, Corradini and Montanari [2,1] for nondeterministic contextual processes – referred to in the definition below is the least preorder in which t precedes t', written $t \nearrow t'$, if either $\partial_1(t) \cap (\partial_0(t') \cup \varsigma(t')) \neq \varnothing$, see Figures 3($a$) and 3($b$), or $\varsigma(t) \cap \partial_0(t') \neq \varnothing$, see Figure 3($c$).

Definition 4. *A (deterministic)* contextual process net *is a finite, acyclic w.r.t.* $\nearrow$*, contextual net* Θ *such that*

1. *for all* $t \in T_\Theta$*,* $\partial_0(t)$ *and* $\partial_1(t)$ *are sets (as opposed to multisets), and*
2. *for all pairs* $t_0 \neq t_1 \in T_\Theta$*,* $\partial_i(t_0) \cap \partial_i(t_1) = \varnothing$*, for* $i = 0,1$.

Remark 1. One could argue that in the contextual process net illustrated in Figure 3(c) the transition t' might also fire before t, inhibiting it. In fact, this cannot be the case. Since the net is a process, i.e., the description of a deterministic run, both t and t' must fired, and the only possible interpretation is that t must execute before t'. There is however no causal dependence between the two events, but only a temporal one. Therefore $t \nearrow t'$ means that t precedes t', either causally or just temporally.

Two transitions t and t' in a deterministic occurrence net are called *concurrent* if they are not related by $\nearrow$ (i.e., if they are not causally or temporally dependent one from the other). We remark that the same definition does not apply to nondeterministic processes, where the concurrency relation must be defined on arbitrary sets of transitions and not just on pairs.

Definition 5. *A* contextual process π *of a contextual net* N *consists of a contextual process net* Θ *together with a pair of functions* $\langle \pi_T, \pi_S \rangle$*, where* $\pi_T : T_\Theta \to T_N$ *and* $\pi_S : S_\Theta \to S_N$*, that respect source, target and context, i.e., such that* $\partial_{N\,i} \circ \pi_T = \pi_S \circ \partial_{\Theta i}$*, for* $i = 0,1$*, and* $\varsigma_N \circ \pi_T = \pi_S \circ \varsigma_\Theta$*. Contextual processes are considered up to isomorphism.*

If no confusion can arise, we denote the components π_T and π_S just by π.

1.2 Petri nets are monoids

The paper [23] built on the monoidal structure of markings to provide an algebraic characterization of the concurrent computations of nets. The basic idea was to lift the structure of states to the level of transitions, providing an algebraic representation of concurrent firing. In turn, these 'algebraic' steps can be sequentially concatenated in order to express more complex computations. While sequential composition endows computations with a categorical structure – markings are objects, computations are arrows, and idle tokens are identities – the parallel composition yields a tensor product. The interplay of parallel and sequential composition, regulated by functoriality of tensor products, models a basic fact about concurrency, namely that concurrent transitions can occur in any relative order. Under the *CTph* the tensor product can simply be commutative. Then, each PT net N freely generates a *strictly* symmetric strict monoidal category $\mathcal{T}(N)$ whose arrows are in bijection with the *commutative processes* of N [3].

Under the *ITph* the situation is more complex. To be able to model causal dependencies, multisets of transitions are not enough. Degano, Meseguer and Montanari proposed to keep simple markings as objects, but to consider a tensor product *non* commutative on the arrows, together with a collection of arrows that may be used to explicitly change the order in which transitions fetch and produce tokens [13]. Such arrows, collected together as the components of a natural isomorphism, turn out to be the classical notion of *symmetry* in category theory, thus leading to the construction of a (non strictly) symmetric strict monoidal category $\mathcal{P}(N)$ for each net N, whose arrows define the *concatenable processes* of N. A more concrete construction, $Q(N)$, was introduced in [32] in order to remove some deficiencies of the previous approach. The main feature of $Q(N)$, which captures the so-called *strongly concatenable processes*, is that its objects are strings rather than multisets of tokens.

For the reader's convenience, we briefly recall the definition of monoidal categories and related concepts. As usual, for $\mathcal{C}$ a category, we denote the identity arrow on the object u by $id_u\colon u \to u$ and the composition of two arrows $f\colon u \to v$ and $g\colon v \to w$ by $f;g\colon u \to w$ (i.e., the operation $\_;\_$ composes in the diagrammatic order). In what follows we let $O_{\mathcal{C}}$ and $A_{\mathcal{C}}$ denote respectively the objects and the arrows of $\mathcal{C}$ and let $\times$ denote the ordinary cartesian product of categories.

Definition 6. *A* strict monoidal category *is a triple $\langle \mathcal{C}, \otimes, e\rangle$, where $\mathcal{C}$ is the underlying category, the functor $\otimes\colon \mathcal{C} \times \mathcal{C} \to \mathcal{C}$ is called* tensor product *and the object $e \in O_{\mathcal{C}}$ is called the* unit. *Moreover, the tensor product satisfies the associativity law $f \otimes (g \otimes h) = (f \otimes g) \otimes h$ for all $f, g, h \in A_{\mathcal{C}}$ and has the constant functor associated to e as neutral element, i.e., $id_e \otimes f = f = f \otimes id_e$, for all $f \in A_{\mathcal{C}}$.*

For non-strict monoidal categories, the associativity and unit laws are satisfied only up to suitable natural isomorphisms. Since we shall always consider strict monoidal categories, the adjective 'strict' can be omitted to simplify the terminology. When the tensor product is commutative up to a suitable natural isomorphism, the monoidal category is called 'symmetric'.

Definition 7. *A* symmetric monoidal category *is a 4-tuple* $\langle \mathcal{C}, \otimes, e, \gamma \rangle$*, where* $\langle \mathcal{C}, \otimes, e \rangle$ *is a monoidal category and* $\gamma\colon \_{1} \otimes \_{2} \Rightarrow \_{2} \otimes \_{1}$ *is a natural isomorphism satisfying the Kelly-MacLane coherence axioms expressed by the following equations:*

$$\begin{aligned} \gamma_{u,v}; \gamma_{v,u} &= id_u \otimes id_v \\ \gamma_{u,v \otimes w} &= (\gamma_{u,v} \otimes id_w); (id_v \otimes \gamma_{u,w}) \end{aligned}$$

for all objects $u, v, w \in O_{\mathcal{C}}$*.*

Note that the equality $\gamma_{u,e} = id_u$ follows from the fact that $u \otimes e = u$ together with axioms above. When γ is the identity natural transformation, then the tensor product is commutative and the category is called 'strictly symmetric'. The arrows of a symmetric monoidal category that can be obtained as the sequential and parallel composition of identities and symmetries are called *permutations* and ranged by σ, σ', σ_1, and so on.

Definition 8. *Let* $\langle \mathcal{C}, \otimes, e \rangle$ *and* $\langle \mathcal{C}', \otimes', e' \rangle$ *be monoidal categories. A functor* $F\colon \mathcal{C} \to \mathcal{C}'$ *is called* strict monoidal *if* $F(e) = e'$ *and* $F(f \otimes g) = F(f) \otimes' F(g)$ *for all* $f, g \in A_{\mathcal{C}}$*.*

Again, we shall omit the term 'strict', since all monoidal functors that we consider are so. The category of monoidal categories and monoidal functors is commonly indicated by **MonCat**. Moreover, we denote by **CMonCat** the full subcategory of strictly symmetric monoidal categories, and use $\mathbf{MonCat}^{\oplus}$ and $\mathbf{CMonCat}^{\oplus}$ for the full subcategory of, respectively, **MonCat** and **CMonCat** consisting of monoidal categories whose sets of objects are freely generated commutative monoids. In particular, we have that both $\mathcal{T}(N)$ and $\mathcal{CT}(N)$ belong to $\mathbf{CMonCat}^{\oplus}$.

Definition 9. *Let* $\langle \mathcal{C}, \otimes, e, \gamma \rangle$ *and* $\langle \mathcal{C}', \otimes', e', \gamma' \rangle$ *be symmetric monoidal categories. A monoidal functor* $F\colon \langle \mathcal{C}, \otimes, e \rangle \to \langle \mathcal{C}', \otimes', e' \rangle$ *is said* symmetric *if* $F(\gamma_{u,v}) = \gamma'_{F(u),F(v)}$*.*

We denote by **SSMC** the subcategory of **MonCat** whose objects are symmetric monoidal categories and whose arrows are symmetric monoidal functors. Let $\mathbf{SSMC}^{\otimes}$ (resp. $\mathbf{SSMC}^{\oplus}$) be the full subcategory of **SSMC** consisting of monoidal categories whose sets of objects are freely generated monoids (resp. commutative monoids). We have $\mathcal{P}(N) \in \mathbf{SSMC}^{\oplus}$ and $\mathcal{Q}(N) \in \mathbf{SSMC}^{\otimes}$.

2 Collective contexts

In [21], Meseguer suggested to represent contexts in rewriting logic theories by considering two kinds of entities for each term: 'counters' and 'copies'. Given a term, one can release as many copies of it as needed, while recording the number of such copies in the corresponding counter. Copies can only be accessed as contexts. On the contrary, when rewriting a term, one has to retrieve the counter and as many copies as indicated by the counter. That is, all the copies ever released. Formally, in the case of contextual nets, the data type of places is modified as follows:

$$\begin{aligned} a &= \langle a, 0 \rangle \\ \langle a, n \rangle &= \langle a, n+1 \rangle \oplus [a]. \end{aligned}$$

$$
\begin{array}{llll}
(1) & (r)^+ \oplus (r)^- = r & (5) & ((r)^-)^- = \varnothing \\
(2) & ((r)^-)^+ = (r)^- & (6) & ((r)^+)^- = (r)^- \\
(3) & (r \oplus s)^+ = (r)^+ \oplus (s)^+ & (7) & (r \oplus s)^- = (r)^- \oplus (s)^- \\
(4) & (\varnothing)^+ = \varnothing & (8) & (\varnothing)^- = \varnothing
\end{array}
$$

Figure 4.

The terms having the form $\langle a,n\rangle$ (for a a place and n a natural number) are counters, and the $[a]$ are copies, with $a = \langle a,n\rangle \oplus n \cdot [a]$. Then, a transition with precondition a, context b and postcondition c becomes a rewrite rule $a \oplus [b] \Rightarrow c \oplus [b]$ with a self-loop on a copy of b. However, this fits well with the *CTph* approach only.

We tried to characterize the algebraic structure that gives the basis for Meseguer's encoding and have come out successfully with a representation that can be extended to deal with the *ITph* as well. As explained in the Introduction, we build the algebraic theory over a non-free monoid of places. In particular, apart from the commutative monoidal operation $\_ \oplus \_$ with unit $\varnothing$, we consider other two operations $(\_)^+$ and $(\_)^-$ that are axiomatized as in Figure 4, where we omit the usual associativity, commutativity and unit axioms for $\_ \oplus \_$. Quite simply, these mean that $(\_)^+$ and $(\_)^-$ are monoid homomorphisms – laws (3), (4), (7) and (8) – such that $(\_)^+ \oplus (\_)^- = id$, $(\_)^+ \circ (\_)^- = (\_)^-$, and $(\_)^- \circ (\_)^- = \varnothing$. Observe that (6) actually follows from (1), (7) and (5). We call *molecules*, ranged over by r, s, $\dots$, the elements of this algebra. Given a set S, we let $\mu(S)$ denote the set of molecules generated by S.

By these laws we can always eliminate consecutive applications of $(\_)^+$ and $(\_)^-$, except for sequences of $(\_)^+$. We shall write r^k as a shorthand for $(\_)^+$ applied k times to r and omit the parentheses. We assume $r^0 = r$, but we remark that in general $r^+ = r^1 \neq r$.

Lemma 1. *For each molecule $r \in \mu(S)$ and each $k \in \mathbb{N}$, we have $(r^k)^- = r^-$.*

Proof. By induction on k, applying law (6).

Proposition 1. *For each molecule $r \in \mu(S)$ and each $k \in \mathbb{N}$, we have $r^k = r^{k+1} \oplus r^-$.*

Proof. By law (1), we have $r^k = (r^k)^+ \oplus (r^k)^-$, and $(r^k)^- = r^-$ by Lemma 1.

Corollary 1. *For each molecule $r \in \mu(S)$ and each $k \in \mathbb{N}$, we have $r = r^k \oplus k \cdot r^-$.*

Of course we are interested in molecules generated from places, which can be of two forms: either a^k or a^-. From the computational point of view, the a^- are the basic contexts and carry very little information, since the nucleus a^k can produce as many of them as needed. To appreciate the point, we can think of the tokens as *ticket rolls* with unbounded number of tickets available. Readers just take a ticket and return it after use for recycle, whereas consumers must retrieve the entire roll, including all used tickets.

Definition 10. *For $N = (S,T,\partial_0,\partial_1,\varsigma)$ a contextual net, define $\mathcal{M}(N)$ as the category in* **CMonCat** *with objects the molecules on S, and arrows generated from the rules in Figure 5, modulo the axioms of strictly symmetric strict monoidal categories in Figure 6.*

$$\frac{r \in \mu(S)}{id_r : r \to r} \qquad \frac{t \in T}{t : \varsigma(t)^{-} \oplus \partial_0(t) \to \varsigma(t)^{-} \oplus \partial_1(t)}$$

$$\frac{\alpha : r \to s,\ \beta : s \to s'}{\alpha ; \beta : r \to s'} \qquad \frac{\alpha : r \to s,\ \beta : r' \to s'}{\alpha \oplus \beta : r \oplus r' \to s \oplus s'}$$

Figure 5.

$$\begin{array}{ccc} \alpha;(\beta;\delta) = (\alpha;\beta);\delta & \alpha; id_s = id_r; \alpha = \alpha & \\ \alpha \oplus (\beta \oplus \delta) = (\alpha \oplus \beta) \oplus \delta & \alpha \oplus \beta = \beta \oplus \alpha & \alpha \oplus id_{\varnothing} = \alpha \\ (\alpha;\beta) \oplus (\delta;\eta) = (\alpha \oplus \delta);(\beta \oplus \eta) & id_{r \oplus s} = id_r \oplus id_s & \end{array}$$

Figure 6.

We can now characterize contextual commutative processes algebraically.

Theorem 1. *The category $\mathcal{CT}(N)$ is isomorphic (via a monoidal functor) to the full subcategory of $\mathcal{M}(N)$ whose objects are $S_N^{\oplus}$.*

A very important property needed in the proof is what we call the *maximum sharing hypothesis*, that can be expressed as in the proposition below. This contains the core of the *CTph* for contextual nets, since it shows that whenever two or more tokens in the same place a are used as contexts, we can always find an equivalent computation where only one token in a is used (twice or more) as a context. In other words, tokens in the same place are completely interchangeable in contexts.

Proposition 2. *For each molecule $r \in \mu(S)$ and $k, n \in \mathbb{N}$, we have $r^n \oplus r^k = r^{n+k} \oplus r$.*

Proof. By Corollary 1, we have $r^{n+k} \oplus r = r^{n+k} \oplus r^k \oplus k \cdot r^{-}$. By commutativity (and associativity) of $\_ \oplus \_$ we get $r^{n+k} \oplus r = r^{n+k} \oplus k \cdot r^{-} \oplus r^k$. By applying k times Proposition 1 we have the result.

Before proving Theorem 1 we need some other technical lemmata.

Lemma 2. *Each molecule $r \in \mu(S)$ factorizes uniquely as $u \oplus e \oplus l$ where*

- ▷ $u \in S^{\oplus}$;
- ▷ $e = k_1 \cdot a_1^{-} \oplus \ldots \oplus k_n \cdot a_n^{-}$ *with $n \geq 0$ and $k_i > 0$, for $i = 1, \ldots, n$;*
- ▷ $l = b_1^{h_1} \oplus \ldots \oplus b_m^{h_m}$ *with $m \geq 0$ and $h_j > 0$, for $j = 1, \ldots, m$;*

where all the a_i and b_j are distinct places.

Proof. The normal form representation follows by observing that $(\_)^{+}$ and $(\_)^{-}$ are monoid homomorphisms and, therefore, distribute over $\oplus$. Then, by laws (2), (5) and (6), we can reduce the molecule to the ‘sum’ of places a, electrons a^{-} and nuclei a^h. Then, by Proposition 2, we can simplify the expression to a form where at most one nucleus a^h with $h > 0$ is present for each a. Finally, if both a^h and $k \cdot a^{-}$ are present in the expression, we can simplify the expression according to the following three possibilities, until all the nuclei and electrons refer to different places.

$(h > k)$: then $a^h = (a^{h-k})^k$ and, by Lemma 1, $k \cdot a^{-} = k \cdot (a^{h-k})^{-}$, hence $a^h \oplus k \cdot a^{-} = (a^{h-k})^k \oplus k \cdot (a^{h-k})^{-} = a^{h-k}$ by Corollary 1;

$(h = k)$: then $a^k \oplus k \cdot a^{-} = a$ by Corollary 1;

$(h < k)$: then $a^h \oplus k \cdot a^{-} = a^h \oplus h \cdot a^{-} \oplus (k-h) \cdot a^{-} = a \oplus (k-h) \cdot a^{-}$ by applying Corollary 1 to $a^h \oplus h \cdot a^{-}$.

Lemma 3. *If the source of an arrow $\alpha \in \mathcal{M}(N)$ factorizes according to Lemma 2 as $u \oplus e \oplus l$, then $\alpha : u \oplus e \oplus l \to v \oplus e \oplus l$ for some $v \in S^{\oplus}$.*

Proof. It is straightforward to observe that e and l are invariants of the generation rules in Figure 5.

Lemma 4. *Each arrow $\delta : r \to s$ in $\mathcal{M}(N)$ can be decomposed as*

$$(t_1 \oplus id_{r_1}); (t_2 \oplus id_{r_2}); \ldots; (t_k \oplus id_{r_k}),$$

for some $k \geq 0$, where all the t_i are transitions.

Proof. By structural induction on the expression denoting α. The complex case is when $\delta = \alpha \otimes \beta$. We can then apply the functoriality of $\oplus$ to get $\delta = (\alpha \oplus id_{r'}); (\beta \oplus id_{s'})$ where r' is the source of β and s' is the target of α. Then we apply the inductive hypothesis to α and β.

We are now ready to prove the main theorem.

Proof. (of Theorem 1). We start by defining the functor $\mathsf{F} : \mathcal{CT}(N) \to \mathcal{M}(N)$. Given a generic step sequence $u_0\,[X_1\rangle\, u_1 \ldots u_{n-1}\,[X_n\rangle\, u_n$ with length n (representing a generic arrow in $\mathcal{CT}(N)$), we let

$$\mathsf{F}(u_0\,[X_1\rangle\, u_1 \ldots u_{n-1}\,[X_n\rangle\, u_n) = \mathsf{F}(u_0\,[X_1\rangle\, u_1); \ldots; \mathsf{F}(u_{n-1}\,[X_n\rangle\, u_n),$$

with $\mathsf{F}(u\,[X\rangle\, v)$ as defined below. Let

- ▹ $u_X = \bigoplus_{t \in \lfloor X \rfloor} X(t) \cdot \partial_0(t)$;
- ▹ $v_X = \bigoplus_{t \in \lfloor X \rfloor} X(t) \cdot \partial_1(t)$;
- ▹ $w_X = \bigoplus_{t \in \lfloor X \rfloor} X(t) \cdot \varsigma(t)$.

We can assume that

$$w_X = k_1 \cdot a_1 \oplus k_2 \cdot a_2 \oplus \ldots \oplus k_m \cdot a_m$$

with $m \geq 0$, $k_i > 0$, for $i = 1, \ldots, m$ and all the a_i different places. Since the step X is enabled at u, then $u = u' \oplus u_X \oplus \lfloor w_X \rfloor$ for some $u' \in S^{\oplus}$. Hence $v = u' \oplus v_X \oplus \lfloor w_X \rfloor$. With this notation fixed, let

$$\mathsf{F}(u\,[X\rangle\, v) = id_{u'} \oplus X \oplus id_{a_1^{k_1}} \oplus id_{a_2^{k_2}} \oplus \ldots \oplus id_{a_m^{k_m}}.$$

Note that with this definition, the k_i tokens needed as context relatively to place a_i yield an idle nucleus $a_i^{k_i}$, for $i = 1, \ldots, m$. Also notice that when $X = \varnothing$ the result is just the identity on u.

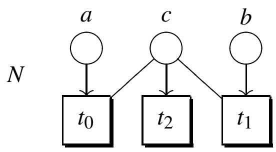

Figure 7.

To show that the mapping F is well-defined we must show that it respects the diamond equivalence, i.e., that when $u\,[X \oplus Y\rangle\, v$ is defined, then

$$\mathsf{F}(u\,[X \oplus Y\rangle\, v) = \mathsf{F}(u\,[X\rangle\, u_1);\mathsf{F}(u_1\,[Y\rangle\, v)$$

with u_1 uniquely determined by u and X. This follows easily by definition of F and by the functoriality of the tensor product.

To show that F is faithful it suffices to observe that the only axiom that potentially may break this property (i.e., that could induce too many equalities on terms) is the functoriality of tensor product which, on the other hand, corresponds precisely to the diamond equivalence.

Finally, to show that F is full (on the full subcategory of $\mathcal{M}(N)$ whose objects are markings), we take a generic arrow $\alpha : u \to v \in \mathcal{M}(N)$ with $u, v \in S^{\oplus}$ and show that there exists a step sequence in $\mathcal{CT}(N)$ that is mapped to α by F. In fact, by Lemma 4 we take a 'linearization' of α (i.e., a sequential composition of transitions in parallel with identities) and show that the obvious firing sequence associated to it can be executed in N. In doing this we employ Lemma 3 and the fact that $u \in S^{\oplus}$. Observe that this construction defines the inverse to F. ♦

Example 1. Let us consider the net N in Figure 7. In $\mathcal{M}(N)$ we have three basic arrows

- ▷ $t_0 : a \oplus c^- \to c^-$,
- ▷ $t_1 : b \oplus c^- \to c^-$ and
- ▷ $t_2 : c \to \varnothing$,

but neither t_0, nor t_1 can represent a commutative contextual process, since their sources and targets are not elements of $S^{\oplus}$. To remedy this, we must put t_0 and t_1 in an environment where the c^- become instances of a 'complete' token, as $id_{c^+} \oplus t_0 : a \oplus c \to c$ and $id_{c^+} \oplus t_1 : b \oplus c \to c$. The concurrent execution of t_0 and t_1 with shared context is instead written as $id_{c^2} \oplus t_0 \oplus t_1 : a \oplus b \oplus c \to \varnothing$: since two electrons are needed the idle nucleus has 'degree' 2. By the functoriality of $\_ \oplus \_$, we have that

$$id_{c^2} \oplus t_0 \oplus t_1 = (id_{c^+} \oplus t_0 \oplus id_b);(id_{c^+} \oplus t_1) = (id_{c^+} \oplus t_1 \oplus id_a);(id_{c^+} \oplus t_0),$$

(recall that $id_{c^2} \oplus id_{c^-} = id_{c^+}$), i.e., t_0 and t_1 can execute in any order. Also interesting is to observe that

$$(id_{c^+} \oplus t_0) \oplus ((id_{c^+} \oplus t_1);t_2) = ((id_{c^+} \oplus t_0);t_2) \oplus (id_{c^+} \oplus t_1),$$

i.e., we have no causal information about the token consumed by t_2: is it the one read by t_0, or the one read by t_1? In fact by $id_{c^+} \oplus t_0 = (id_{c^+} \oplus t_0); id_c$ and applying the functoriality of $\oplus$ we have:

$$(id_{c^+} \oplus t_0) \oplus ((id_{c^+} \oplus t_1); t_2) = (id_{c^+} \oplus t_0 \oplus id_{c^+} \oplus t_1); (id_c \oplus t_2).$$

Then, $id_c \oplus t_2 = t_2 \oplus id_c$ by commutativity of $\oplus$ and by applying the functoriality (in the opposite direction than before) we get the equality. Furthermore,

$$id_{c^+} \oplus t_0 \oplus id_{c^+} \oplus t_1 = id_{c^2} \oplus t_0 \oplus t_1 \oplus id_c$$

(by Proposition 2 and the commutativity of $\oplus$), and thus

$$(id_{c^+} \oplus t_0) \oplus ((id_{c^+} \oplus t_1); t_2) = (id_{c^2} \oplus t_0 \oplus t_1 \oplus id_c); (id_c \oplus t_2) = id_{c^2} \oplus t_0 \oplus t_1 \oplus t_2$$

i.e., t_0, t_1, and t_2 can be executed in a concurrent fashion without the possibility of distinguishing this case from those in which t_2 causally depends on t_0 or t_1.

Since a PT net N can always be seen as a contextual net with no read-arcs, in which case the commutative contextual processes of N are just the ordinary commutative processes (cf. § 1.2 and [23]), then by Theorem 1 we obtain the following corollary.

Corollary 2. *If N is a* PT *net, then $\mathcal{T}(N)$ is a full subcategory of $\mathcal{M}(N)$.*

We remark that the constructions we have shown can be easily extended to deal with multiplicities on read-arcs (i.e., to the case in which $\varsigma(t)$ is a multiset rather than a set).

3 Individual contexts

The maximum sharing hypothesis creates obvious problems when dealing with the *ITph*, whose entire point is to be able to recognize what electrons are emitted from each token. For ordinary PT nets, the information about causality is recovered in the algebraic setting by using (non strictly) symmetric strict monoidal categories, i.e., by introducing *symmetries* to control rearrangements of tokens in process sequential composition. While at the level of states one can still view standard markings as indexed collections of ordered tokens (rather than resorting to take as states the elements of the free monoid on places, i.e., strings of places), at the level of computations (arrows), however, the tensor product is *not* commutative anymore, so that one is able to interpret in a canonical way the correct flow of causality through token histories. Thus, the first attempt to a uniform extension of the *CTph* treatment of the previous section to the *ITph* view is to introduce symmetries on molecules.

There is however another problem to solve. Since the context $\varsigma(t)$ is modeled by a self-loop on $\varsigma(t)^-$, two transitions with the same context can be concatenated on it, as if one depended on the execution of the other. This spurious causal dependency is to be avoided, as it gives rise to a wrong semantic model. To some extent, one would like to follow the *ITph* on 'complete' molecules (standard markings), and the *CTph* on electrons of the same nucleus, so that one has no information about which electron is

(9)	$a^+ \otimes a^- = a$	(13)	$(p^-)^- = \varnothing$
(10)	$(p^-)^+ = p^-$	(14)	$(p^+)^- = p^-$
(11)	$(p \otimes q)^\chi = p^\chi \otimes q^\chi$	(15)	$\varnothing^\chi = \varnothing$
(12)	$a^\delta \otimes b^\varepsilon = b^\varepsilon \otimes a^\delta$	(16)	$a \otimes a^- = a^- \otimes a$

Figure 8. Axioms for bimolecules (with $a \neq b \in S$, $\delta, \varepsilon \in \mathbb{N} \cup \{^-\}$ and $\chi \in \{^+, ^-\}$).

consumed by a firing, but only about which molecule it comes from. We therefore need a canonical interpretation of molecules that respects this intuition. To fix the ideas, we take initially a non commutative monoidal operation $\otimes$ on molecules. Let us consider the molecule $a^2 \otimes a^+ \otimes a^- \otimes a^- \otimes a^-$. We would like to view it under an interpretation that connects each of the three electrons with one the two nuclei, and that is *invariant* not only under all possible computations that can originate from the state, but also under composition of the molecule to form larger states. Our idea is to associate an electron to the *first* incomplete nucleus (*ion*) that *precedes* it. In the present case, for instance, the first electron is associated to the second ion (a^+), while the second and third electrons (the two rightmost in the expression) are interpreted as electrons released by the leftmost ion a^2. A good way to explain the mechanism, is to view ions as open parentheses and electrons as closing parentheses, where of course an ion a^k opens several parentheses, namely k, at once. Clearly, we are mainly interested in balanced expressions, but unbalanced expressions must exist, and can always be completed by parallel composition to yield balanced terms. To complete the picture, consider now that the order in which atoms, nuclei and electrons of different kinds – i.e., coming from different places – appear in an expression is not relevant. Hence, the monoidal operation $\otimes$ better be commutative in such situations. In other words, we have:

$$a^+ \otimes b^+ \otimes a^- \otimes b^- = a^+ \otimes b^+ \otimes b^- \otimes a^- = a^+ \otimes b \otimes a^- = a^+ \otimes a^- \otimes b = a \otimes b,$$

but we definitely want that $a^+ \otimes a^+ \otimes a^- \neq a^+ \otimes a^- \otimes a^+$, because the particle a^- in the two terms is associated to different nuclei and, therefore, the two states may give rise to different causal histories when a transition reads that particle.

We call *bimolecules*, ranged over by $p, q, \ldots$, the (generalized) markings of the algebra illustrated above. It includes a set of axioms formally identical to those in Figure 4, plus some extra axioms to deal with restricted commutativity. Given a set S, we write $\nu(S)$ for the set of bimolecules on S. The complete axiomatization of bimolecules is shown in Figure 8. Note that law (9) – the analogous of (1) for molecules for bimolecules applies only when a nucleus is immediately on the left of an electron, i.e., $a^- \otimes a^+ \neq a$.

The final and key ingredient in our construction is to *abandon* the symmetry of the monoidal categories involved. With a step similar to the one that brought from strictly symmetric to symmetric categories, we choose (*non* symmetric) monoidal categories to which we adjoin *exactly* and *only* the symmetries we need. In this way, we are able to omit those symmetries that would cause migration of electrons from atom to atom. In the following we shall build on a construction somehow intermediate between $\mathcal{P}(N)$ and

$$\frac{p \in \nu(S)}{id_p\colon p \to p} \qquad \frac{t \in T}{t\colon \varsigma(t)^- \otimes \partial_0(t) \to \varsigma(t)^- \otimes \partial_1(t)} \qquad \frac{u,v \in S^\otimes}{\gamma_{u,v}\colon u \otimes v \to v \otimes u}$$

$$\frac{\alpha\colon p \to q,\ \beta\colon q \to r}{\alpha;\beta\colon p \to r} \qquad \frac{\alpha\colon p \to q,\ \beta\colon p' \to q'}{\alpha \otimes \beta\colon p \otimes p' \to q \otimes q'}$$

Figure 9.

$$\begin{array}{lll}
\alpha;(\beta;\sigma)=(\alpha;\beta);\sigma & \alpha;id_q=id_p;\alpha=\alpha & (\alpha;\beta)\otimes(\alpha';\beta')=(\alpha\otimes\alpha');(\beta\otimes\beta') \\
\alpha\otimes(\beta\otimes\sigma)=(\alpha\otimes\beta)\otimes\sigma & \alpha\otimes id_\varnothing=id_\varnothing\otimes\alpha=\alpha & id_{p\otimes q}=id_p\otimes id_q \\
(\alpha\otimes\beta);\gamma_{q,q'}=\gamma_{p,p'};(\beta\otimes\alpha) & \gamma_{p,q};\gamma_{q,p}=id_p\otimes id_q & \gamma_{p,q\otimes r}=(\gamma_{p,q}\otimes id_r);(id_q\otimes\gamma_{p,r})
\end{array}$$

Figure 10.

$Q(N)$ for PT nets [12,32] and, therefore, take a non commutative monoid of objects: it is commutative only on some objects, in particular on the markings. We use the symbol $\otimes$ for the monoidal operation.

Definition 11. *For $N = (S,T,\partial_0,\partial_1,\varsigma)$ a contextual net, $\mathcal{B}(N)$ is the monoidal category with objects the bimolecules on S, and arrows generated from the rules in Figure 9, together with the symmetries*

$$\begin{aligned}
\gamma_{a^\delta,b^\varepsilon} &: a^\delta \otimes b^\varepsilon \to b^\varepsilon \otimes a^\delta, \quad \text{for } a \neq b \in S \text{ and } \delta,\varepsilon \in \mathbb{N} \cup \{^-\}, \\
\gamma_{a,a^-} &: a \otimes a^- \to a^- \otimes a, \\
\gamma_{a^-,a} &: a^- \otimes a \to a \otimes a^-.
\end{aligned}$$

The arrows are taken modulo the axioms of strict monoidal categories in Figure 10 (whenever the γ's are defined) and the laws:

$$\sigma;t;\sigma' = t \tag{17}$$

$$\gamma_{a^\delta,b^\varepsilon} = id_{a^\delta \otimes b^\varepsilon}, \quad \text{for } a \neq b \in S \text{ and } \delta,\varepsilon \in \mathbb{N} \cup \{^-\} \tag{18}$$

$$\gamma_{a,a^-} = id_{a \otimes a^-} \tag{19}$$

$$id_{a^k} \otimes t = id_{a^{k+1}} \otimes t \otimes id_{a^-} \tag{Δ}$$

for all transitions $t\colon p \to q$, permutations $\sigma\colon p \to p$, $\sigma'\colon q \to q$, and $k > 0$.

Since $\gamma_{a^-,a}$ is inverse to $\gamma_{a,a^-} = id_{a\otimes a^-}$, it follows that $\gamma_{a^-,a} = id_{a\otimes a^-}$. Note that we *do not* introduce symmetries such as γ_{a^-,a^-}, γ_{a^k,a^-}, and γ_{a^k,a^n}, for $k,n \geq 1$, that would allow the particles to flow from a nucleus to a different one. For example, starting from $a \otimes a^+ = a^+ \otimes a^- \otimes a^+$ and applying an hypothetical arrow $id_{a^+} \otimes \gamma_{a^-,a^+}$, we would reach $a^+ \otimes a^+ \otimes a^- = a^+ \otimes a$, allowing the nuclei to exchange electrons, which is problematic. Another *non*-example would be applying the arrow $id_{a^+\otimes a^+} \otimes \gamma_{a^-,a^-}$ to $a \otimes a = a^+ \otimes a \otimes a^- = a^+ \otimes a^+ \otimes a^- \otimes a^-$ because, after the exchange, the token of the first and second nucleus get confused. By forcing γ_{a^-,a^-} to be the identity we would confuse the electrons of two different nuclei, because of the naturality axiom, and by leaving it free we would allow again for electrons migration. In fact, our *representation invariant* is that the electrons associated to a certain nucleus a^k in a bimolecule q are determined by following the discipline of proper nesting of open and closed parentheses. The absence of those symmetries maintains this invariant for us.

Laws (17) and (18) are classical laws for the $\mathcal{P}(N)$ construction; here they have a slightly more general role, because they also deal with nuclei and electrons. In particular, law (17) is the analogous of axiom (Ψ) for PT nets (cf. [12]). Law (19) says that electrons can be freely moved around 'complete atoms' of the same kind. Law (Δ) is original and really central to our development. In fact, even though the symmetries γ_{a^-,a^-} are not allowed, we certainly do not want to distinguish between electrons of the same nucleus (the first released, the second, ...), as otherwise we would obtain a notion of computation very concrete and far from our target, that is to capture algebraically the notion of contextual process. Axiom (Δ) takes care of identifying such particles, as the Example 2 below illustrates.

To establish our representation result we need to refine contextual processes in order to be able to concatenate them. As for similar cases in the literature, this leads to the introduction of an *ordering* on the tokens in the source and target of the process net, yielding the notion of *concatenable contextual processes*.

Definition 12. *For N a contextual net, a* concatenable contextual process *is a tuple $(\pi, \Theta, \prec_0, \prec_1)$, where π is a contextual process with underlying contextual process net Θ, $\prec_0$ and $\prec_1$ are partial orders on the minimal and maximal places of Θ, respectively, such that: (1) $x \prec_i y$ implies that $\pi(x) = \pi(y)$; and (2) if $x \neq y$ are minimal places (respectively maximal places) such that $\pi(x) = \pi(y)$, then either $x \prec_0 y$ or $y \prec_0 x$ (respectively, $x \prec_1 y$ or $y \prec_1 x$).*

As usual, concatenable processes are taken up to isomorphism. The two conditions imposed in the definition above ensure that we order only places of Θ that are instances of the same place of N, and that on such places the ordering is total.

Example 2. Let N be the contextual net in Figure 7. Then we have three basic arrow in $\mathcal{B}(N)$ associated to the transitions of N:

- $t_0 : c^- \otimes a \to c^-$;
- $t_1 : c^- \otimes b \to c^-$;
- $t_2 : c \to \varnothing$.

Then, new arrows can be built by composing (sequentially and in parallel) these three arrows with identities and symmetries. For example, the arrow $id_{c^2} \otimes t_0 \otimes t_1$ goes from $c^2 \otimes c^- \otimes a \otimes c^- \otimes b = c \otimes a \otimes b$ to $c^2 \otimes c^- \otimes c^- = c$. Analogously we have the arrow $id_{c^2} \otimes t_1 \otimes t_0 : c \otimes b \otimes a \to c$. Then, it is possible to prove that these two arrows are identified in $\mathcal{B}(N)$. In fact, we have:

$$\begin{aligned} id_{c^2} \otimes t_0 \otimes t_1 &= (id_{c^2} \otimes t_0 \otimes id_{c^- \otimes b}); (id_{c^2} \otimes id_{c^-} \otimes t_1) \text{ (by functoriality)} \\ &= (id_{c^+} \otimes t_0 \otimes id_b); (id_{c^+} \otimes t_1) \text{ (by law } \Delta). \end{aligned}$$

Then, by naturality, we have $t_0 \otimes id_b = \gamma_{c^- \otimes a,b}; (id_b \otimes t_0); \gamma_{b,c^-}$, but these symmetries are just identities and therefore $t_0 \otimes id_b$ can be replaced by $id_b \otimes t_0$ in the expression above.

$$\begin{aligned} id_{c^2} \otimes t_0 \otimes t_1 &= (id_{c^+} \otimes id_b \otimes t_0); (id_{c^+} \otimes t_1) \\ &= (id_{c^2 \otimes c^-} \otimes id_b \otimes t_0); (id_{c^2} \otimes t_1 \otimes id_{c^-}) \text{ (by law } \Delta) \\ &= id_{c^2} \otimes t_1 \otimes t_0 \text{ (by functoriality)} \end{aligned}$$

Note that in fact there is only one concatenable contextual process that starts from $a \otimes b \otimes c$ and involves exactly one firing of t_0 and one firing of t_1. By repeatedly applying law (Δ) we then have, e.g.,

$$id_{c^2} \otimes t_0 \otimes t_1 = id_{c^{n+m+2}} \otimes t_1 \otimes id_{n \cdot c^-} \otimes t_0 \otimes id_{m \cdot c^-}$$

for all $n,m \in \mathbb{N}$. This means that the order in which the electrons are read is not important provided that they originated from the same nucleus.

Likewise concatenable processes of PT nets, a partial operation of *sequential composition* can be defined. Provided the target of process π coincides with the source of process π', it merges the *maximal* places of π with the *minimal* places of π' according to the orders $\prec_1$ and $\prec'_0$.

Definition 13. *Let $(\pi, \Theta, \prec_0, \prec_1)$ and $(\pi', \Theta', \prec'_0, \prec'_1)$ be two concatenable contextual processes of a contextual net N, where $T_\Theta \cap T_{\Theta'} = \varnothing$ and $S_\Theta \cap S_{\Theta'}$ is both the set of minimal places for Θ and the set of maximal places for Θ', with $\pi(x) = \pi'(x)$ for any $x \in S_\Theta \cap S_{\Theta'}$, and $x \prec_1 y$ iff $x \prec'_0 y$ for all $x,y \in S_\Theta \cap S_{\Theta'}$. Then their concatenation $(\pi'', \Theta \cup \Theta', \prec_0, \prec'_1) = (\pi, \Theta, \prec_0, \prec_1); (\pi', \Theta', \prec'_0, \prec'_1)$ is well defined, where π'' is the componentwise union of π and π' (i.e., $\pi''(x) = \pi(x)$ if $x \in \Theta$ and $\pi''(x) = \pi'(x)$ if $x \in \Theta'$).*

The composition is well defined because by hypothesis we have $\pi'(x) = \pi(x)$ for all $x \in \Theta \cap \Theta' = S_\Theta \cap S_{\Theta'}$, i.e., merged places have the same names.

The *parallel composition* of two processes consists of taking their disjoint union and extending the orders on minimal and maximal places by $x \prec_i y$ whenever x belongs to the first process, y to the second, and $\pi(x) = \pi(y)$. It can be shown that with these two operations the concatenable contextual processes of N form the arrows of a *strict monoidal category* $\mathcal{CP}(N)$. *Symmetries* can be defined by taking a process that contains just places (no transitions) with suitable orderings $\prec_0$ and $\prec_1$. Each place is both minimal and maximal. These symmetries make $\mathcal{CP}(N)$ be a symmetric monoidal category in $\mathbf{SSMC}^\oplus$.

Definition 14. *A concatenable contextual process is called* elementary *if it contains at most one transition.*

Definition 15. *Given a contextual net N and a transition $t \in T_N$, the elementary concatenable contextual process $[t] = (\pi, \Theta, \prec_0, \prec_1)$ associated to t is given by*

▷ $S_\Theta = \{\langle a,0,n \rangle \mid a \in \lfloor \partial_0(t) \rfloor,\ 1 \le n \le \partial_0(t)(a)\} \cup \{\langle a,1,n \rangle \mid a \in \lfloor \partial_1(t) \rfloor,\ 1 \le n \le \partial_1(t)(a)\} \cup \{\langle a,2,1 \rangle \mid a \in \lfloor \varsigma(t) \rfloor\}$

▷ $T_\Theta = \{\langle t\rangle\}$;
▷ $\partial_0(\langle t\rangle) = \{\langle a,0,n\rangle \mid a \in \lfloor\partial_0(t)\rfloor,\ 1 \leq n \leq \partial_0(t)(a)\}$;
▷ $\partial_1(\langle t\rangle) = \{\langle a,1,n\rangle \mid a \in \lfloor\partial_1(t)\rfloor,\ 1 \leq n \leq \partial_1(t)(a)\}$;
▷ $\varsigma(\langle t\rangle) = \{\langle a,2,1\rangle \mid a \in \lfloor\varsigma(t)\rfloor\}$;
▷ $\pi(\langle a,j,n\rangle) = a$ *and* $\pi(\langle t\rangle) = t$;
▷ $\langle a,i,k\rangle \prec_1 \langle b,j,h\rangle$ *iff* $a = b \wedge i = j = 0 \wedge k < h$.
▷ $\langle a,i,k\rangle \prec_1 \langle b,j,h\rangle$ *iff* $a = b \wedge i = j = 1 \wedge k < h$.

Note that the places in $\{\langle a,2,1\rangle \mid a \in \lfloor\varsigma(t)\rfloor\}$ are both minimal and maximal. No order is needed on them, because we rely on the basic assumptions that $\varsigma(t)$ is a set and that $\varsigma(t) \cap \lfloor\partial_0(t) \cup \partial_1(t)\rfloor = \varnothing$, for any $t \in T_N$.

Proposition 3. *Each elementary concatenable contextual process* $(\pi,\Theta,\prec_0,\prec_1)$ *that contains exactly one transition, say x, can be obtained as* $\sigma_1;([\pi(x)]\otimes\sigma_2);\sigma_3$ *for suitable elementary concatenable contextual processes* σ_1, σ_2 *and* σ_3 *that contains no transition.*

Proposition 4. *The concatenable contextual processes of a contextual net N can be obtained as the sequential composition of elementary concatenable contextual processes.*

Proof. Likewise the analogous statement for ordinary PT nets, the proof is by induction on the number of transitions in the process net (exploiting Proposition 3).

Theorem 2. *The category* $\mathcal{CP}(N)$ *is isomorphic (via a symmetric monoidal functor) to the full subcategory of* $\mathcal{B}(N)$ *whose objects are the elements of* $S^\otimes$ *(which is symmetric).*

Before proving the main representation theorem above, we need some technical lemmata that state useful properties of the arrows in $\mathcal{B}(N)$. We start by extending some of the properties of molecules to the framework of bimolecules.

Lemma 5. *For each bimolecule p and each* $k \in \mathbb{N}$*, we have* $(p^k)^- = p^-$.

Proposition 5. *For each place a and each* $k \in \mathbb{N}$ *, we have* $a^k = a^{k+1} \otimes a^-$.

Proof. The proof proceeds by induction on k. For the base case ($k = 0$) we get $a = a^+ \otimes a^-$ directly by law (9). For the inductive case, we assume the property to be valid for $k = n$ and prove it for $k = n+1$. Then,

$$\begin{aligned} a^{n+1} &= (a^n)^+ \text{ (by definition)} \\ &= (a^{n+1} \otimes a^-)^+ \text{ (by inductive hypothesis)} \\ &= (a^{n+1})^+ \otimes (a^-)^+ \text{ (by law 11)} \\ &= a^{n+2} \otimes a^- \text{ (by law 10).} \end{aligned}$$

This concludes the proof.

Corollary 3. *For each place a and each* $k \in \mathbb{N}$*, we have* $a = a^k \otimes k \cdot a^-$.

Lemma 6. *Each bimolecule p can be decomposed as* $p = p_1 \otimes p_2 \otimes \ldots \otimes p_n$*, where each* p_i *has the form* $k_{i,0} \cdot a_i^- \otimes a_i^{k_{i,1}} \otimes a_i^{k_{i,2}} \otimes \ldots \otimes a_i^{k_{i,n_i}}$ *with* $a_i \neq a_j$*, for* $i \neq j$.

Lemma 7. *If $p \otimes q \in S^{\otimes}$, then for each $u \in S^{\otimes}$ we have $p \otimes u \otimes q \in S^{\otimes}$.*

Proof. It suffices to prove the property for $u \in S$, which can be done via a simple case analysis, exploiting the representation of p and q provided by Lemma 6 and applying law (16).

Note that in the previous lemma, p and q are generic bimolecules and not necessarily markings.

We can now state some invariant and decomposition properties for the arrows in $\mathcal{B}(N)$.

Lemma 8. *If $\alpha = id_p \otimes \gamma_{x,y} \otimes id_q$ with $p \otimes x \otimes y \otimes q \in S^{\otimes}$, then $p \otimes y \otimes x \otimes q \in S^{\otimes}$*

Proof. By a simple case analysis: all symmetries are collapsed to identities, except when $x = y = a$ for some $a \in S$.

Lemma 9. *If $\alpha = id_p \otimes t \otimes id_q$ and $p \otimes \varsigma(t)^{-} \otimes \partial_0(t) \otimes q \in S^{\otimes}$, then $p \otimes \varsigma(t)^{-} \otimes \partial_1(t) \otimes q \in S^{\otimes}$.*

Proof. Follows from Lemma 7.

Proposition 6. *Each $\alpha \in \mathcal{B}(N)$ can be decomposed as*

$$\alpha = \sigma_0; (id_{p_1} \otimes t_1 \otimes id_{q_1}); \sigma_1; (id_{p_2} \otimes t_2 \otimes id_{q_2}); \sigma_2; \ldots; (id_{p_n} \otimes t_n \otimes id_{q_n}); \sigma_n,$$

where the σ_i are permutations (i.e., sequential and parallel compositions of symmetries and identities) and the t_i are transitions.

Proof. By structural induction. The complex case is for $\alpha = \alpha_1 \otimes \alpha_2$ for some $\alpha_1 : r_1 \to r_1'$ and $\alpha_2 : r_2 \to r_2'$. But then, by functoriality we have $\alpha = (\alpha_1 \otimes id_{r_2}); (id_{r_1'} \otimes \alpha_2)$ and by inductive hypothesis

$$\begin{aligned}
\alpha_1 &= \sigma_0'; (id_{p_1'} \otimes t_1' \otimes id_{q_1'}); \sigma_1'; (id_{p_2'} \otimes t_2' \otimes id_{q_2'}); \sigma_2'; \ldots; (id_{p_{n'}'} \otimes t_{n'}' \otimes id_{q_{n'}'}); \sigma_{n'}', \\
\alpha_2 &= \sigma_0''; (id_{p_1''} \otimes t_1'' \otimes id_{q_1''}); \sigma_1''; (id_{p_2''} \otimes t_2'' \otimes id_{q_2''}); \sigma_2''; \ldots; (id_{p_{n''}''} \otimes t_{n''}'' \otimes id_{q_{n''}''}); \sigma_{n''}'',
\end{aligned}$$

Then, by functoriality:

$$\begin{aligned}
\alpha_1 \otimes id_{r_2} &= (\sigma_0' \otimes id_{r_2}); (id_{p_1'} \otimes t_1' \otimes id_{q_1' \otimes r_2}); (\sigma_1' \otimes id_{r_2}); (id_{p_2'} \otimes t_2' \otimes id_{q_2' \otimes r_2}); \\
&\quad (\sigma_2' \otimes id_{r_2}); \ldots; (id_{p_{n'}'} \otimes t_{n'}' \otimes id_{q_{n'}' \otimes r_2}); (\sigma_{n'}' \otimes id_{r_2}), \\
id_{r_1'} \otimes \alpha_2 &= (id_{r_1'} \otimes \sigma_0''); (id_{r_1' \otimes p_1''} \otimes t_1'' \otimes id_{q_1''}); (id_{r_1'} \otimes \sigma_1''); (id_{r_1' \otimes p_2''} \otimes t_2'' \otimes id_{q_2''}); \\
&\quad (id_{r_1'} \otimes \sigma_2''); \ldots; (id_{r_1' \otimes p_{n''}''} \otimes t_{n''}'' \otimes id_{q_{n''}''}); (id_{r_1'} \otimes \sigma_{n''}''),
\end{aligned}$$

From which the hypothesis follows trivially – $\sigma_i = \sigma_i' \otimes id_{r_2}$, for $i = 0, \ldots, n'-1$, $\sigma_{n'} = (\sigma_{n'}' \otimes id_{r_2}); (id_{r_1'} \otimes \sigma_0'')$, and $\sigma_{n'+i} = id_{r_1'} \otimes \sigma_i''$, for $i = 1, \ldots, n''$. ♦

The main law (Δ) can then be extended to generic arrows whenever we know that the rightmost electron belongs to the nucleus that precedes the arrow.

Corollary 4. *For each* $\alpha: p \to q \in \mathcal{B}(N)$*,* $a \in S$ *and* $k > 0$ *such that* $a^{k+1} \otimes p \otimes a^{-} = a^k \otimes p$ *we have* $id_{a^{k+1}} \otimes \alpha \otimes id_{a^-} = id_{a^k} \otimes \alpha$*.*

Corollary 5. *If* $\alpha: u \to q \in \mathcal{B}(N)$ *with* $u \in S^{\otimes}$*, then* $q \in S^{\otimes}$*.*

Proof. Consequence of Proposition 6 and Lemmata 8 and 9.

Lemma 10. *If* $\alpha = id_p \otimes t \otimes id_q \in \mathcal{B}(N)$ *with* t *a transition and* $p \otimes \varsigma(t)^{-} \otimes \partial_0(t) \otimes q \in S^{\otimes}$*, then* $\alpha = \sigma; (id_{\varsigma(t)^+} \otimes t \otimes id_u); \sigma'$ *for some permutations* σ *and* σ' *and some marking* $u \in S^{\otimes}$*.*

Proof. By the decomposition of Lemma 6, and by the fact that the source of α is a marking, it follows that $p = \bigotimes_{a \in S} a^{k_{a,1}} \otimes \ldots \otimes a^{k_{a,n_a}}$ and $q = \bigotimes_{a \in S} h_a \cdot a^{-} \otimes h'_a \cdot a$. It follows that each electron a^{-} in $\varsigma(t)^{-}$ belongs to the closest ion on the left of the electron (namely, the i_ath nucleus of type a in p with i_a the greatest index in $1 \le i_a \le n_a$ such that $k_{a,i_a} > 0$). Moreover, the h_a electrons of type a in q can be attached to their corresponding nuclei in p, by applying law (Δ). Therefore we have $\alpha = id_{p'} \otimes t \otimes id_v$ where $p' = \bigotimes_{a \in S}(i_a - 1) \cdot a \otimes a^{+} \otimes (n_a - i_a) \cdot a$ (if a is not read by t then the corresponding argument in the sum is just $n_a \cdot a$), and $v = \bigotimes_{a \in S} h'_a \cdot a$. Then, by naturality of symmetries, we have:

$$\alpha = (id_{p''} \otimes \gamma_{v', \varsigma(t)^{-} \otimes \partial_0(t)} \otimes id_v); (id_{p''} \otimes t \otimes id_{v' \otimes v}); (id_{p''} \otimes \gamma_{\varsigma(t)^{-} \otimes \partial_1(t), v'} \otimes id_v),$$

where $p'' = \bigotimes_{a \in S}(i_a - 1) \cdot a \otimes a^{+}$ and $v' = \bigotimes_{a \in S}(n_a - i_a) \cdot a$. In fact the symmetries that we have used in the expression are defined since they involve the swappings of 'complete' tokens with either 'complete' tokens, or electrons. By naturality we have also:

$$(id_{p''} \otimes t \otimes id_{v' \otimes v}) = (\gamma_{v'', \varsigma(t) \otimes \partial_0(t)} \otimes id_{v' \otimes v}); (id_{\varsigma(t)^+} \otimes t \otimes id_{v'' \otimes v' \otimes v}); (\gamma_{\varsigma(t) \otimes \partial_1(t), v''} \otimes id_{v' \otimes v})$$

where $v'' = \bigotimes_{a \in S}(i_a - 1) \cdot a$. By taking

$$\begin{aligned} \sigma &= (id_{p''} \otimes \gamma_{v', \varsigma(t)^{-} \otimes \partial_0(t)} \otimes id_v); (\gamma_{v'', \varsigma(t) \otimes \partial_0(t)} \otimes id_{v' \otimes v}) \\ \sigma' &= (\gamma_{\varsigma(t) \otimes \partial_1(t), v''} \otimes id_{v' \otimes v}); (id_{p''} \otimes \gamma_{\varsigma(t)^{-} \otimes \partial_1(t), v'} \otimes id_v) \\ u &= v'' \otimes v' \otimes v \end{aligned}$$

we have the thesis. ♦

Proposition 7. *Each* $\alpha: u \to q \in \mathcal{B}(N)$ *with* $u \in S^{\otimes}$ *can be decomposed as*

$$\alpha = \sigma_0; (id_{\varsigma(t_1)^+} \otimes t_1 \otimes id_{u_1}); \sigma_1; (id_{\varsigma(t_2)^+} \otimes t_2 \otimes id_{u_2}); \sigma_2; \ldots; (id_{\varsigma(t_n)^+} \otimes t_n \otimes id_{u_n}); \sigma_n,$$

where the σ_i *are permutations, the* t_i *are transitions and* $u_i \in S^{\otimes}$*, for* $i = 1, \ldots, n$*.*

Proof. The proof exploits the decomposition provided by Proposition 6 and then applies n times the result of Lemma 10.

We are now ready to prove the main representation result of this section.

Proof. (of Theorem 2). We start by defining the monoidal functor $\mathsf{G}\colon \mathcal{CP}(N) \to \mathcal{B}(N)$, which is the identity on objects. By Proposition 3, the functor is completely determined by defining the mapping of elementary processes, since then $\mathsf{G}(\alpha;\beta) = \mathsf{G}(\alpha);\mathsf{G}(\beta)$ and $\mathsf{G}(\alpha \otimes \beta) = \mathsf{G}(\alpha) \otimes \mathsf{G}(\beta)$. For symmetries, the mapping is the classical one (see e.g. [31]). For the elementary process $[t]$ associated to the transition $t \in T_N$, we let $\mathsf{G}([t]) = id_{\varsigma(t)^+} \otimes t$. It remains to prove that:

1. G is well defined;
2. G is full (on the full subcategory of $\mathcal{B}(N)$ whose objects are markings);
3. G is faithful.

The fact that G is well defined means that different decompositions of the same process in terms of elementary processes are mapped to the same arrow. This corresponds to show that different orderings of the events in a process $\sigma = (\pi, \Theta, \prec_0, \prec_1)$ that are consistent with the ordering of events $\nearrow_\Theta$ yield the same arrow in $\mathcal{B}(N)$. To see this, it suffices to show that given a decomposition of the process σ, and taken any two concurrent events that are executed consecutively according to the order imposed by the fixed decomposition, then the decomposition in which the two concurrent events are executed in the reverse order is mapped to the same arrow of σ. The proof is easy (by functoriality of the tensor product) if the two events do not share a context. Otherwise, axiom (Δ) must be employed, as we did in Example 2. Formally, we consider the process $P = P_1;([t_1] \otimes \sigma_1);\sigma;([t_2] \otimes \sigma_2);P_2$ where σ_1 is the identity process on the marking $u_2 \oplus \partial_0(t_2) \oplus v$, σ is the process associated to the permutation $id_u \otimes \gamma_{u_1 \otimes \partial_1(t_1), u_2 \oplus \partial_0(t_2)} \otimes id_v$, and σ_2 is the identity process on the marking $u_1 \oplus \partial_1(t_2) \oplus v$, i.e., $\varsigma(t_1) = u \oplus u_1$, $\varsigma(t_2) = u \oplus u_2$, and the two occurrences share the context u (note that while u_1 and u_2 are not necessarily disjoint, the corresponding sets of tokens read by t_1 and t_2 in the process P are disjoint). Then, we have also $P = P_1;\sigma';([t_2] \otimes \sigma_2');\sigma'';([t_1] \otimes \sigma_1');\sigma''';P_2$, for suitable permutation processes:

$$
\begin{array}{lll}
\sigma' & \text{associated to} & id_u \otimes \gamma_{u_1 \oplus \partial_0(t_1), u_2 \oplus \partial_0(t_2)} \otimes id_v, \\
\sigma_2' & \text{idle process associated to} & u_1 \oplus \partial_0(t_1) \oplus v, \\
\sigma'' & \text{associated to} & id_u \otimes \gamma_{u_2 \oplus \partial_1(t_2), u_1 \oplus \partial_0(t_1)} \otimes id_v, \\
\sigma_1' & \text{idle process associated to} & u_2 \oplus \partial_1(t_2) \oplus v, \\
\sigma''' & \text{associated to} & id_u \otimes \gamma_{u_1 \oplus \partial_1(t_1), u_2 \oplus \partial_1(t_2)} \otimes id_v.
\end{array}
$$

Hence we want to prove that the two decompositions are mapped to the same arrow in $\mathcal{B}(N)$. More precisely, we show that

$$\mathsf{G}(([t_1] \otimes \sigma_1);\sigma;([t_2] \otimes \sigma_2)) = \mathsf{G}(\sigma';([t_2] \otimes \sigma_2');\sigma'';([t_1] \otimes \sigma_1');\sigma''').$$

The complete proof is shown in Figure 11. We briefly comment the critical steps:

Step 20: we have exploited axiom (Δ) and then the fact that symmetries on electrons and tokens are identities to transform the second subexpression;

Step 21: we have applied the naturality of symmetries to the first and second subexpressions – in order to match source and target of t_1 with the components of the symmetries, observe that $u_1 \otimes u^- = u_1^+ \otimes u^- \otimes u_1^-$ since u and u_1 are disjoint;

$$
\begin{aligned}
&\mathsf{G}(([t_1]\otimes\sigma_1);\sigma;([t_2]\otimes\sigma_2)) = \\
&= (id_{u^+\otimes u_1^+}\otimes t_1\otimes id_{u_2\otimes\partial_0(t_2)\otimes v});(id_u\otimes\gamma_{u_1\otimes\partial_1(t_1),u_2\otimes\partial_0(t_2)}\otimes id_v);(id_{u^+\otimes u_2^+}\otimes t_2\otimes id_{u_1\otimes\partial_1(t_1)\otimes v}) \\
&= (id_{u^+\otimes u_1^+}\otimes t_1\otimes id_{u_2\otimes\partial_0(t_2)\otimes v});(id_{u^+}\otimes\gamma_{u_1\otimes u^-\otimes\partial_1(t_1),u_2\otimes\partial_0(t_2)}\otimes id_v); \\
&\quad (id_{u^+\otimes u_2^+}\otimes t_2\otimes id_{u_1\otimes\partial_1(t_1)\otimes v}) && (20) \\
&= (id_{u^+}\otimes\gamma_{u_1\otimes u^-\otimes\partial_0(t_1),u_2\otimes\partial_0(t_2)}\otimes id_v);(id_{u^+\otimes u_2\otimes\partial_0(t_2)\otimes u_1^+}\otimes t_1\otimes id_v); \\
&\quad (id_{u^+\otimes u_2^+}\otimes t_2\otimes id_{u_1\otimes\partial_1(t_1)\otimes v}) && (21) \\
&= (id_{u^+}\otimes\gamma_{u_1\otimes u^-\otimes\partial_0(t_1),u_2\otimes\partial_0(t_2)}\otimes id_v);(id_{u^2\otimes u_2^+\otimes u^-\otimes u_2^-\otimes\partial_0(t_2)\otimes u_1^+}\otimes t_1\otimes id_v); \\
&\quad (id_{u^2\otimes u_2^+}\otimes t_2\otimes id_{u_1^+\otimes u^-\otimes u_1^-\otimes\partial_1(t_1)\otimes v}) && (22) \\
&= (id_{u^+}\otimes\gamma_{u_1\otimes u^-\otimes\partial_0(t_1),u_2\otimes\partial_0(t_2)}\otimes id_v);(id_{u^2\otimes u_2^+}\otimes t_2\otimes id_{u_1^+}\otimes t_1\otimes id_v); && (23) \\
&= (id_{u^+}\otimes\gamma_{u_1\otimes u^-\otimes\partial_0(t_1),u_2\otimes\partial_0(t_2)}\otimes id_v);(id_{u^2\otimes u_2^+}\otimes t_2\otimes id_{u_1^+\otimes u^-\otimes u_1^-\otimes\partial_0(t_1)\otimes v}); \\
&\quad (id_{u^2\otimes u_2^+\otimes u^-\otimes u_2^-\otimes\partial_1(t_2)\otimes u_1^+}\otimes t_1\otimes id_v); && (24) \\
&= (id_{u^+}\otimes\gamma_{u_1\otimes u^-\otimes\partial_0(t_1),u_2\otimes\partial_0(t_2)}\otimes id_v);(id_{u^+\otimes u_2^+}\otimes t_2\otimes id_{u_1\otimes\partial_0(t_1)\otimes v}); \\
&\quad (id_{u^+\otimes u_2\otimes\partial_1(t_2)\otimes u_1^+}\otimes t_1\otimes id_v); && (25) \\
&= (id_{u^+}\otimes\gamma_{u_1\otimes u^-\otimes\partial_0(t_1),u_2\otimes\partial_0(t_2)}\otimes id_v);(id_{u^+\otimes u_2^+}\otimes t_2\otimes id_{u_1\otimes\partial_0(t_1)\otimes v}); \\
&\quad (id_{u^+}\otimes\gamma_{u_2\otimes\partial_1(t_2),u_1\otimes u^-\otimes\partial_0(t_1)}\otimes id_v);(id_{u^+\otimes u_1^+}\otimes t_1\otimes id_{u_2\otimes\partial_1(t_2)\otimes v}); \\
&\quad (id_{u^+}\otimes\gamma_{u_1\otimes u^-\otimes\partial_1(t_1),u_2\otimes\partial_1(t_2)}\otimes id_v); && (26) \\
&= (id_u\otimes\gamma_{u_1\otimes\partial_0(t_1),u_2\otimes\partial_0(t_2)}\otimes id_v);(id_{u^+\otimes u_2^+}\otimes t_2\otimes id_{u_1\otimes\partial_0(t_1)\otimes v}); \\
&\quad (id_u\otimes\gamma_{u_2\otimes\partial_1(t_2),u_1\otimes\partial_0(t_1)}\otimes id_v);(id_{u^+\otimes u_1^+}\otimes t_1\otimes id_{u_2\otimes\partial_1(t_2)\otimes v}); \\
&\quad (id_u\otimes\gamma_{u_1\otimes\partial_1(t_1),u_2\otimes\partial_1(t_2)}\otimes id_v); && (27) \\
&= \mathsf{G}(\sigma';([t_2]\otimes\sigma_2');\sigma'';([t_1]\otimes\sigma_1'))
\end{aligned}
$$

Figure 11. The proof of $\mathsf{G}(([t_1]\otimes\sigma_1);\sigma;([t_2]\otimes\sigma_2)) = \mathsf{G}(\sigma';([t_2]\otimes\sigma_2');\sigma'';([t_1]\otimes\sigma_1'))$.

Step 22: we have used axiom (Δ) to transform the second and third subexpressions;
Step 23: we have applied the functoriality of the tensor product to the second and third subexpressions;
Step 24: we have applied the functoriality of the tensor product to the second subexpressions to reverse the order in which t_2 and t_1 appear in the previous expressions;
Step 25: we have used axiom (Δ) to reduce the second and third subexpressions;
Step 26: we have applied the naturality of symmetries twice to expand the third subexpression;
Step 27: we have used axiom (Δ) and then the fact that symmetries on electrons are identities to transform the first, third and fifth subexpressions.

The fact that G is full follows from Propositions 4 and 7, since $\mathsf{G}([t]) = id_{\varsigma(t)^+}\otimes t$.

Finally, regarding faithfulness, let P_0 and P_1 be such that $\mathsf{G}(P_0) = \mathsf{G}(P_1)$, and let α be a term representing $\mathsf{G}(P_0)$. Observe, by simply inspecting the axioms that define $\mathcal{B}(N)$, that all the possible choices for α have the same number of transitions. More precisely, exactly the same transitions occur in each term obtained by rewriting α according to such axioms. Moreover, by definition of G, these are in one-to-one correspondence with the transitions of P_0 and with those of P_1. We can therefore proceed by induction on the number n of transitions of α (and P_0 and P_1) to prove that P_0 and P_1 are isomorphic processes.

The base case, where n equals zero, is obvious, as α is simply a permutation. For the induction case, let fix any decomposition of α according to Proposition 7, say

$$\alpha = \sigma_0; (id_{\varsigma(t_1)^+} \otimes t_1 \otimes id_{u_1}); \sigma_1; (id_{\varsigma(t_2)^+} \otimes t_2 \otimes id_{u_2}); \sigma_2; \ldots; (id_{\varsigma(t_n)^+} \otimes t_n \otimes id_{u_n}); \sigma_n,$$

An argument similar to the one employed to establish the well-definedness of G, but working in the opposite direction, proves that all the steps needed to transform α in the normal form selected above can be mimicked both on P_0 and P_1. It then follows that P_i, for $i = 0,1$, can be written as $P_i = P_i'; \sigma_i; ([t_n] \otimes \sigma_i'); \sigma_i''$, where $\mathsf{G}(P_0') = \mathsf{G}(P_1')$. Then, by induction hypothesis, we can conclude that P_0' and P_1' are isomorphic processes. It is then easy to prove that so are P_0 and P_1. ♦

Besides the fact that *all* the arrows of $\mathcal{B}(N)$ have a meaningful computational interpretation, a further advantage of the present approach with respect to the match-share categories of [16] is that the arrows of the model category corresponding to pure concatenable process can be distinguished *just* by looking at their sources and targets, rather than by inspecting their construction. And as for the *CTph*, our proposal is a conservative extension of the ordinary concatenable process semantics (cf. § 1.2 and [12,31]).

Corollary 6. *If N is a* PT *net, then $\mathcal{P}(N)$ is a full subcategory of $\mathcal{B}(N)$.*

Moreover, the present axiomatics of $\mathcal{B}(N)$ improves sensibly the construction presented in [7]. In particular, the monoid of objects is here 'morally' commutative, thus making redundant the idea of instances of transitions and the related axioms [32,7]. Moreover, the exact sharing hypothesis has found a mature, satisfactory formulation in terms of law (Δ) which, among other things, allowed us to dispense with the particles a_-.

Concluding Remarks and Future Work

Building on an important suggestion of Meseguer in [21], we have shown a way to extend the algebraic semantics of PT nets proposed in [23] to contextual nets, both in the collective token and the individual token interpretation. The constructions rely on the choice of a non-free monoid of objects, whose elements we called molecules and bimolecules. In the case of the collective token philosophy, our work extends Meseguer's by identifying the *maximum sharing hypothesis* as the fundamental law of collective contextual processes. The key to transport these ideas to the individual token philosophy was to renounce to the symmetry of the monoidal category, being thus able to select only the symmetries consistent with our computational interpretation in terms of concatenable contextual processes. The axioms of *exact sharing* provided us with a way to regulate the interplay between all the different ingredients.

Although we have worked only at the level of single nets, we believe that our approach can be extended to constructions between categories of nets and models, with restrictions analogous to those well-known in the literature [31,32].

Acknowledgements. We would like to thank José Meseguer and Paolo Baldan for some interesting discussion on the topic and also Matteo Coccia for his reading of a preliminary version of our work.

References

1. P. Baldan. Modelling concurrent computations: From contextual Petri nets to graph grammars. Ph.D. thesis, TD-1/00, Dipartimento di Informatica, Università di Pisa, 2000.
2. P. Baldan, A. Corradini, and U. Montanari. An event structure semantics for P/T contextual nets: Asymmetric event structures. In *Proc. FoSSaCS'98, Foundations of Software Science and Computation Structures*, (M. Nivat, Ed.), vol. 1378 of *Lect. Notes in Comput. Sci.*, pp. 63–80. Springer, 1998.
3. E. Best and R. Devillers. Sequential and concurrent behaviour in Petri net theory. *Theoretical Computer Science*, 55:87–136, 1987.
4. R. Bruni. Tile Logic for Synchronized Rewriting of Concurrent Systems. Ph.D. thesis, TD-1/99, Dipartimento di Informatica, Università di Pisa, 1999.
5. R. Bruni, J. Meseguer, U. Montanari, and V. Sassone. A comparison of Petri net semantics under the collective token philosophy. In *Proc. ASIAN'98, 4th Asian Computing Science Conference*, (J. Hsiang, A. Ohori, Eds.), vol. 1538 of *Lect. Notes in Comput. Sci.*, pp. 225–244. Springer, 1998.
6. R. Bruni, J. Meseguer, U. Montanari, and V. Sassone. Functorial semantics for Petri nets under the individual token philosophy. In *Proc. CTCS'99, 8th conference on Category Theory and Computer Science*, (M. Hofmann, G. Rosolini, D. Pavlovic, Eds.), vol. 29 of *Elect. Notes in Comput. Sci.*, 19 pages. Elsevier Science, 1999.
7. R. Bruni and V. Sassone. Algebraic models for contextual nets. In *ICALP2000, 27th Int. Coll. on Automata, Languages and Programming*, (U. Montanari, J. Rolim, E. Welzl, Eds.), vol. 1853 of *Lect. Notes in Comput. Sci.*, pp. 175–186. Springer, 2000.
8. N. Busi and M. Pinna. Non sequential semantics for contextual P/T nets. In *Application and Theory of Petri Nets*, vol. 1091 of *Lect. Notes in Comput. Sci.*, pp. 113–132. Springer, 1996.
9. N. Busi. Petri nets with inhibitor and read arcs: Semantics, analysis and application to process calculi. Ph.D. thesis, TD-1/99, Dipartimento di Informatica, Università di Siena, 1998.
10. S. Christensen and N.D. Hansen. Coloured Petri nets extended with place capacities, test arcs and inhibitor arcs. In *ICATPN'93, 14th Int. Conf. Applications and Theory of Petri Nets*, (M.A. Marsan, Ed.), vol. 691 of *Lect. Notes in Comput. Sci.*, pp. 186–205. Springer, 1993.
11. A. Corradini and U. Montanari. An algebraic semantics for structured transition systems and its application to logic programs. *Theoretical Computer Science*, 103:51–106, 1992.
12. P. Degano, J. Meseguer, and U. Montanari. Axiomatizing net computations and processes. In *Proc. LICS'89, 4th Symposium on Logic in Computer Science*, pp. 175–185. IEEE Computer Society Press, 1989.
13. P. Degano, J. Meseguer, and U. Montanari. Axiomatizing the algebra of net computations and processes. *Acta Inform.*, 33(7):641–667, 1996.
14. J. Desel, G. Juhás, and R. Lorenz. Process semantics of Petri nets over partial algebra. In *Proc. ICATPN 2000, 21st Int. Conf. on Application and Theory of Petri Nets*, (M. Nielsen, D. Simpson, Eds.), vol. 1825 of *Lect. Notes in Comput. Sci.*, pp. 146–165. Springer, 2000.

15. H. Ehrig and J. Padberg. Uniform approach to Petri nets. In *Proc. Foundations of Computer Science: Potential-Theory-Cognition*, (C. Freska, M. Jantzen, R. Valk, Eds.), vol. 1337 of *Lect. Notes in Comput. Sci.*, pp. 219–231. Springer, 1997.
16. F. Gadducci and U. Montanari. Axioms for contextual net processes. In *Proc. ICALP'98, 25th International Colloquium on Automata, Languages, and Programming*, (K.G. Larsen, S. Skyum, G. Winskel, Eds.), vol. 1443 of *Lect. Notes in Comput. Sci.*, pp. 296–308. Springer, 1998.
17. F. Gadducci and U. Montanari. The tile model. In *Proof, Language and Interaction: Essays in Honour of Robin Milner*, (G. Plotkin, C. Stirling, and M. Tofte, Eds.). MIT Press, 2000.
18. U. Goltz and W. Reisig. The non-sequential behaviour of Petri nets. *Inform. and Comput.*, 57:125–147, 1983.
19. R. Janicki and M. Koutny. Semantics of inhibitor nets. *Inform. and Comput.*, 123:1–16, 1995.
20. J. Meseguer, Conditional rewriting logic as a unified model of concurrency, *Theoretical Computer Science*, 96:73–155, 1992.
21. J. Meseguer. Rewriting logic as a semantic framework for concurrency: A progress report. In *Proc. CONCUR'96, 7th International Conference on Concurrency Theory*, (U. Montanari, V. Sassone, Eds.), vol. 1119 of *Lect. Notes in Comput. Sci.*, pp. 331–372. Springer, 1996.
22. J. Meseguer and U. Montanari. Petri nets are monoids: A new algebraic foundation for net theory. In *Proc. LICS'89, 3rd Symposium on Logic in Computer Science*, pp. 155–164. IEEE Computer Society Press, 1988.
23. J. Meseguer and U. Montanari. Petri nets are monoids. *Inform. and Comput.*, 88(2):105–155, 1990.
24. J. Meseguer, U. Montanari, and V. Sassone. On the semantics of place/transition Petri nets. *Math. Struct. in Computer Science*, 7:359–397, 1997.
25. J. Meseguer, P.C. Ölveczky, and M.-O. Stehr. Rewriting logic as a unifying framework for Petri nets. Submitted to this Volume.
26. U. Montanari and F. Rossi. Contextual occurrence nets and concurrent constraint programming. In *Graph Transformations in Computer Science*, vol. 776 of *Lect. Notes in Comput. Sci.*, pp. 280–285. Springer, 1994.
27. U. Montanari and F. Rossi. Contextual nets. *Acta Inform.*, 32:545–596, 1995.
28. C.A. Petri. *Kommunikation mit Automaten*. Ph.D. thesis, Institut für Instrumentelle Mathematik, Bonn, 1962.
29. W. Reisig. *Petri Nets: An Introduction*. EACTS Monographs on Theoretical Computer Science. Springer, 1985.
30. G. Ristori. *Modelling Systems with Shared Resources via Petri Nets*. Ph.D. thesis, TD-5/94, Dipartimento di Informatica, Università di Pisa, 1994.
31. V. Sassone. An axiomatization of the algebra of Petri net concatenable processes. *Theoretical Computer Science*, 170:277–296, 1996.
32. V. Sassone. An axiomatization of the category of Petri net computations. *Math. Struct. in Computer Science*, 8:117–151, 1998.
33. R.J. van Glabbeek and G.D. Plotkin. Configuration structures. In *Proc. LICS'95, 10th Symposium on Logic in Computer Science*, pp. 199–209. IEEE Computer Society Press, 1995.
34. W. Vogler. Efficiency of asynchronous systems and read arcs in Petri nets. In *Proc. ICALP'97, 24th International Colloquium on Automata, Languages, and Programming*, vol. 1256 of *Lect. Notes in Comput. Sci.*, pp. 538–548. Springer, 1997.
35. W. Vogler. Partial order semantics and read arcs. In *Proc. MFCS'97, 22nd Int. Symp. on Mathematical Foundations of Computer Science*, (P. Degano, R. Gorrieri, A. Marchetti-Spaccamela, Eds.), vol. 1295 of *Lect. Notes in Comput. Sci.*, pp. 508–517. Springer, 1997.
36. G. Winskel. Event structures. In *Proc. of Advanced Course on Petri Nets*, vol. 255 of *Lect. Notes in Comput. Sci.*, pp. 325–392. Springer, 1986.