Algebraic Models 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 extend the algebraic approach of Meseguer and Montanari from ordinary place/transition Petri nets to *contextual* nets, covering both the *collective* and the *individual* token philosophy *uniformly* along the two interpretations of net behaviors.

Introduction

Among the models for concurrency, *place/transition Petri nets* (PT nets), introduced by Petri in [14] (see also [15]), are one of the most largely diffused, with many interdisciplinary applications. The reasons of the success of the net model probably reside in the simple formal description and natural characterization of *concurrent* and *distributed systems*: the state of a system consists of a (multi)set of distributed resources, its actions consume some of the resources available and release fresh resources, thus affecting only local subsystems. In particular, a computation can be described as a partial order of events such that two events in the same computation 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.

Several extensions of the basic 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 *contextual nets*, also known as nets with *read arcs*, or *condition arcs*, or *test arcs* [4,13,8,21]. The underlying idea is that of *reading* resources *without consuming* them, thus providing a way of modeling multiple concurrent accesses to the same resource. With ordinary PT nets such readings must be rendered as *self-loops*, and this imposes an unfortunate 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 *not* affected by the firing. Contextual nets have found applications e.g., to transaction serializability in databases [16], concurrent constraint programming [12], and asynchronous systems [20].

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. [19]).

Research supported by CNR Integrated Project *Metodi per la Verifica di Sistemi Eterogenei*; by Esprit Working Groups *CONFER2* and *COORDINA*; and by MURST project *TOSCA*.

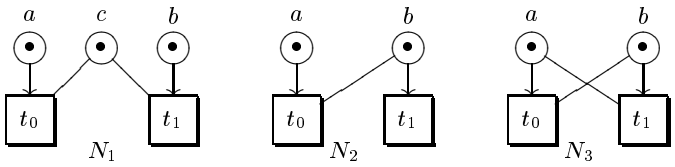

Figure 1.

According to the collective token philosophy (*CTph*), one should not distinguish among different *tokens* in the same place (i.e., among instances of the same resource), because all such tokens are operationally equivalent. This view disregards that tokens may have different origins and histories and may, therefore, carry different *causality* information. Selecting one instance rather than another, can make the difference from being causally dependent or not on some previous event. And this may well be a piece of information one does not want to discard, which is the point of the individual token philosophy (*ITph*). Of course, causal dependencies may influence the degree of concurrency in computations, and therefore *CTph* and *ITph* lead to different concurrent semantics.

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 [21]. (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 [13], 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 [8], instead, the concurrent step is allowed for all three nets, the basic assumption being that t_0 and t_1 can both start together and read the context tokens, without needing them 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 [21] 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 $\{t_0, t_1\}$. 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. In this paper, we follow the interpretation of [13] that fits better our understanding of contexts.

Collecting Tokens. The seminal paper [10] proposed an algebraic approach to the analysis of net behaviors relying on the basic observation that the monoidal structure of PT net states (i.e., the *markings*) can be lifted to the level of computations so to obtain an algebraic initial model for concurrent net behaviors according to the *CTph*.

The algebraic net theory developed under the *CTph* is well consolidated, and the relationships between its *computational*, *algebraic* and *logical* interpretations are by now very clear [3]. Starting with the classical 'token-game' semantics, many behavioral models for Petri nets have been proposed that follow the *CTph*. In particular, the *commutative processes* of Best and Devillers [2] reconcile the 'diamond' equivalence 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 the elements of the free commutative monoid over the set of places, its arrows correspond to the commutative processes of N [10,5].

Surprisingly, the *CTph* semantics for contextual nets have received poor attention in the literature. 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 remove 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 [9], we give here a fully 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* that can emit several 'negative' *particles* a^-, while keeping track of the number of electrons around, i.e., as in [9], we assume that for all $k \in \mathbb{N}$, $a = a^k \otimes k \cdot a^-$, with a^k a shorthand for $a^{+\ldots+}$ (+ applied k times).

Replacing context arcs on a with self-loop arcs on 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*.

Observing Causal Dependencies. Building on the notion of *process* introduced by Goltz and Reisig in [7], several authors have shown that the semantics of nets in the *ITph* can still be understood in terms of *symmetric monoidal categories*. In particular, a simple variation of Goltz-Reisig processes called *concatenable processes* is introduced in [5] (see also [17]), which admits sequential composition and yields a symmetric monoidal category $\mathcal{P}(N)$ for each net N. Also several unfolding semantics (see e.g. [22,11]) 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 [13,1], 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 Monta-

nari [6] using *match-share* categories. There, the basic idea is that, together with symmetries, two additional auxiliary constructors must be present: one for *duplicating* tokens and one for *matching* them. Read arcs can then be replaced by self-loops, 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. Multiple concurrent access is achieved by producing via duplication – and then absorbing via matching – enough copies of the context. In [6] 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. The main problem 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 [6], by noticing that 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: The resulting arrow is meaningless from the computational viewpoint. We overcome this problem by extending to the ITph the approach proposed in the first part of the paper for the CTph. In particular, besides a^+ and a^- we introduce the term a_- for each place a, with $a = a^k \otimes k \cdot a_-$.

Each context arc from a to t is then replaced by putting a^- in the source of t and a_- in the target of t. This is necessary to avoid that contexts released by a transition be consumed by another transition, and represents, in the ITph, a sort of dual to the maximum sharing hypothesis. Then, we introduce symmetries on markings, but *regulate* their use on the a^+, a^- and a_- as to *forbid* the swapping of a a^+ and an adjacent a^- or a_-. This is actually the key of our proposal, as it prevents that electrons may migrate from atom to atom, which is essentially what happens in [6]. We impose this restriction by omitting the corresponding symmetries. Putting such arrows back in the model would in fact result in a redundant framework perfectly analogous to the one of match-share categories. Our main result is that, again, the arrows between standard markings are in bijection with a slight refinement of contextual processes, called *strongly concatenable*.

Structure of the Paper. 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 strongly 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 [4,13,8,21]. 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 $u\colon S \to \mathbb{N}$, where $u(a)$ denotes the number of tokens that place a carries in u. The set of finite multiset on S if a free commutative monoid on S. We denote it by $S^{\oplus}$, and indicate multiset inclusion, difference and union by $\subseteq$, $\oplus$ and $\ominus$, respectively. 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, $\partial_0, \partial_1\colon T \to S^{\oplus}$ are the pre and post-set functions, and $\varsigma\colon T \to S^{\oplus}$ is the context function.*

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. 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*.

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., $\left\lfloor \bigoplus_{t\in T} \varsigma(t) \right\rfloor \oplus \bigoplus_{t\in T} X(t)\cdot\partial_0(t) \subseteq u$, 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 of transitions 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 concurrent point of view, 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 of* contextual commutative processes *$\mathcal{CCP}(N)$ has the markings of N as objects and its step sequences, taken modulo the diamond equivalence, as arrows.*

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.

Definition 4. *A* contextual process net *is a finite, acyclic (w.r.t. the preorder in which t precedes t' if either $\partial_1(t) \cap (\partial_0(t') \cup \varsigma(t')) \neq \varnothing$ or $\varsigma(t) \cap \partial_0(t') \neq \varnothing$) 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 $i = 0, 1$.*

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 isomorphisms.*

1.2 Petri nets are monoids

The paper [10] exploited 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. Since 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 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 [2].

Under the *ITph* the situation is more complex. In order to be able to model causal dependencies one cannot consider multisets of transitions. The proposal of Degano, Meseguer and Montanari was to introduce a non commutative tensor product – while keeping markings as objects – together with suitable arrows for exchanging the order in which transitions fetch and produce tokens [5]. Such arrows are called symmetries, and are formalized categorically as the components of a natural isomorphism. This approach leads 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, $\mathcal{Q}(N)$, was introduced in [18] in order to remove some deficiencies of the previous approach. The main feature of $\mathcal{Q}(N)$, which captures the so-called *strongly concatenable processes*, is that its objects are strings rather than multisets of tokens.

$$
\begin{array}{lll}
(1) & (u)^{+} \oplus (u)^{-}=u & (5) \quad ((u)^{-})^{-}=\varnothing \\
(2) & ((u)^{-})^{+}=(u)^{-} & (6) \quad ((u)^{+})^{-}=(u)^{-} \\
(3) & (u \oplus v)^{+}=(u)^{+} \oplus (v)^{+} & (7) \quad (u \oplus v)^{-}=(u)^{-} \oplus (v)^{-} \\
(4) & (\varnothing)^{+}=\varnothing & (8) \quad (\varnothing)^{-}=\varnothing
\end{array}
$$

Figure 2.

2 Collective contexts

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 2, where we omit the usual associativity, commutativity and unit axioms for $_ \oplus _$. These mean precisely that $(_)^{+}$ and $(_)^{-}$ are monoid homomorphisms such that $(_)^{+} \oplus (_)^{-} = id$, $(_)^{+} \circ (_)^{-} = (_)^{-}$, and $(_)^{-} \circ (_)^{-} = \varnothing$. Observe that (6) actually follows from (1), (7) and (5).

By these laws we can always eliminate consecutive applications of $(_)^{+}$ and $(_)^{-}$, except for sequences of $(_)^{+}$. We shall write u^k as a shorthand for $(_)^{+}$ applied k times to u and omit the parentheses. We assume $u^0 = u$, but we remark that in general $u^{+} = u^1 \neq u$. We call *molecules* the elements of this algebra. Given a set S, we let $\mu(S)$ denote the set of molecules generated by S.

Lemma 1. *For any natural number k and molecule u we have $(u^k)^{-} = u^{-}$.*

Proof. By induction, applying law (6).

Proposition 1. *For any natural k, and molecule u, we have $u^k = u^{k+1} \oplus u^{-}$*

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

Corollary 1. *For any natural k and molecule u, we have $u = u^k \oplus k \cdot u^{-}$.*

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

Definition 6. *Given a contextual net $N = (S, T, \partial_0, \partial_1, \varsigma)$, we define the category $\mathcal{M}(N)$ as the category with objects the molecules on S and with arrows generated from the rules in Figure 3, modulo the axioms of strictly symmetric strict monoidal categories in Figure 4.*

We can now characterize contextual commutative processes algebraically.

$$\frac{u \in \mu(S)}{id_u \colon u \to u} \qquad \frac{t \in T,\ \partial_0(t) = u,\ \partial_1(t) = v,\ \varsigma(t) = w}{t \colon u \oplus w^{-} \to v \oplus w^{-}}$$

$$\frac{\alpha \colon u \to v,\ \beta \colon v \to w}{\alpha;\beta \colon u \to w} \qquad \frac{\alpha \colon u \to v,\ \beta \colon w \to z}{\alpha \oplus \beta \colon u \oplus w \to v \oplus z}$$

Figure 3.

$$\begin{array}{lll} \alpha;(\beta;\gamma)=(\alpha;\beta);\gamma & \alpha;id_v=id_u;\alpha=\alpha & \\ \alpha\oplus(\beta\oplus\gamma)=(\alpha\oplus\beta)\oplus\gamma & \alpha\oplus\beta=\beta\oplus\alpha & \alpha\oplus id_{\varnothing}=\alpha \\ (\alpha;\beta)\oplus(\gamma;\delta)=(\alpha\oplus\gamma);(\beta\oplus\delta) & id_{u\oplus v}=id_u\oplus id_v & \end{array}$$

Figure 4.

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

A very important property needed in the proof is what we call the *maximum sharing hypothesis*, that can be expressed as below. This result contains the core of the *CTph*, 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 is used (twice or more) as context.

Proposition 2. *For any molecule u and natural numbers k and n, we have $u^n \oplus u^k = u^{n+k} \oplus u$.*

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

For instance, let us consider the net N in Figure 5. In $\mathcal{M}(N)$ we have three basic arrows $t_0 \colon a \oplus c^{-} \to c^{-}$, $t_1 \colon b \oplus c^{-} \to c^{-}$ and $t_2 \colon c \to \varnothing$, 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 $id_{c^{+}} \oplus t_0 \colon a \oplus c \to c$ and $id_{c^{+}} \oplus t_1$. The concurrent execution of t_0 and t_1 with shared context is instead written as $id_{c^2} \oplus t_0 \oplus t_1$. 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?

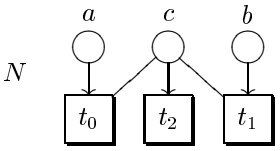

Figure 5.

3 Individual contexts

The maximum sharing hypothesis creates obvious problems when dealing with the *ITph*, whose entire point is to be able to recognize how electrons are emitted from tokens. 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* for controlling rearrangements of tokens when composing processes. At the level of states we still have standard markings. At the level of computations (arrows), however, the tensor product is *not* commutative anymore, so that we are able to interpret correctly the flow of causality through token histories. Thus, the first attempt to a uniform extension of the *CTph* treatment of the previous section 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 $\varsigma(t)^{\text{-}}$, 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. We thus choose to introduce a new kind of electrons, denoted by $u_{\text{-}}$ for representing used (i.e., read) contexts. A transition t consume a *forward copy* of its context $\varsigma(t)^{\text{-}}$ and produces a *backward copy* $\varsigma(t)_{\text{-}}$ that cannot be read by other transitions. We call *bimolecules* the (generalized) markings of the algebra that includes also the operator $(_)_{\text{-}}$ subject to a set of axioms formally identical to those involving $(_)^{\text{-}}$ in Figure 2. Given a set S, we write $\nu(S)$ for the set of bimolecules on S.

The final and key ingredient in our construction is to *abandon* the symmetry of the monoidal categories involved. In 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 the construction $\mathcal{Q}(N)$ for PT nets and, therefore, take a non commutative monoid of objects. We use the symbol $\otimes$ for the monoidal operation, which essentially amounts to string concatenation. Given a string q, we denote by $\mu(q)$ its underlying multiset.

Definition 7. *Given a contextual net $N = (S, T, \partial_0, \partial_1, \varsigma)$, we define the category $\mathcal{B}(N)$ as the category with objects the bimolecules on S and with arrows generated from the rules in Figure 6, together with the symmetries $\gamma_{a^\chi, b^\kappa} : a^\chi \otimes b^\kappa \to b^\kappa \otimes a^\chi$ and $\gamma_{a^\delta, c^\epsilon} : a^\delta \otimes c^\epsilon \to c^\epsilon \otimes a^\delta$, for $a, b, c \in S$ with $a \neq b$, for $\chi, \kappa \in \mathbb{N} \cup \{^{\text{-}}, _{\text{-}}\}$, and for $\delta, \epsilon \in \{^{\text{-}}, _{\text{-}}\}$. The arrows are taken modulo the axioms of strict monoidal categories in Figure 7 (whenever the γ's are defined) and the laws:*

$$s; t_{p,q}; s' = t_{p',q'} \tag{9}$$

$$\gamma_{a^\delta, a^\delta} = id_{a^\delta \otimes a^\delta}, \text{ for } \delta \in \{^{\text{-}}, _{\text{-}}\} \tag{10}$$

for any symmetries $s : p' \to p$ and $s' : q \to q'$, and any transition $t : \mu(p) \to \mu(q)$.

Note that we do *not* introduce the symmetries $\gamma_{a^k, a^{\text{-}}}$ and $\gamma_{a^k, a_{\text{-}}}$ that would allow the electrons to flow from a nucleus to a different one. For example, starting from $a \otimes a = a^+ \otimes a_{\text{-}} \otimes a^+ \otimes a^{\text{-}}$ and applying the arrow $a^+ \otimes \gamma_{a_{\text{-}}, a^+} \otimes a^{\text{-}}$, we would reach

$$\frac{p \in \nu(S)}{id_p : p \to p} \qquad \frac{t \in T,\ \partial_0(t) \oplus \varsigma(t)^{\text{-}} = \mu(p),\ \partial_1(t) \oplus \varsigma(t)_{\text{-}} = \mu(q)}{t_{p,q} : p \to q} \qquad \frac{p, q \in S^{\otimes}}{\gamma_{p,q} : p \otimes q \to q \otimes p}$$

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

Figure 6.

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

Figure 7.

$a^{\text{+}} \otimes a^{\text{+}} \otimes a_{\text{-}} \otimes a^{\text{-}} = a^{\text{+}} \otimes a \otimes a^{\text{-}}$, which is problematic. In fact, our *representation invariant* is that the electrons associated to a certain nucleus a^k in a string q are the first k electrons (either $a^{\text{-}}$ or $a_{\text{-}}$) that appear in q to the right of a^k. Thus, for consistency, we want exactly k electrons between a^k and the successive nucleus a^n occurring in q. The absence of those symmetries maintains this invariant.

As for $\mathcal{Q}(N)$ in [18], we introduce an arrow $t_{p,q}$ for all the possible linearizations p and q of the source and target of each transition t of N. Law (9), considered originally in [18], establishes a link between all the instances of a single t, guaranteeing both consistency and a sensible computational interpretation for such arrows. Actually, (9) expresses that the collection of the instances of t forms a *natural transformation* between suitable functors. The reader is referred to [18] for a thorough discussion of this topic. Laws (10) make the instances of electrons associated to the same nucleus indistinguishable from each other by asserting that the order in they are used is immaterial.

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 introduce an *ordering* of the tokens in the source and target of the process net, yielding the notion of *strongly concatenable contextual processes.*

Definition 8. *Given a net N, a* strongly 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 total orders on the minimal and maximal places of Θ, respectively, such that $a \prec_0 b$ (resp. $a \prec_1 b$) implies $\pi_S(a) = \pi_S(b)$.*

Likewise concatenable processes, 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 $\prec'_0$. The *parallel composition* of two processes consists of taking their disjoint union and extending the orders on minimal and maximal places by taking $a \prec_i b$ whenever a belongs to the first process and b to the second. It can be shown that with these two operations the strongly

concatenable contextual processes of N form the arrows of a *strict monoidal category* $\mathcal{SCCP}(N)$. *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 $\mathcal{SCCP}(N)$ be a symmetric monoidal category. Due to space limitation we cannot give more details here. We refer to [18] for the presentation of the category of strongly concatenable processes which is similar. We can now state the main result of the paper.

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

The proof is quite long and requires the introduction and the description of the algebra of further kinds of processes that represent those arrows whose sources and targets involve nuclei and electrons. In particular, we use suitable inscriptions inside minimal and maximal places in order to represent the electrons which have been moved away from the nuclei of their atoms and their movements. However, such inscriptions are vacuous for processes whose source and target are strings of places (as all the electrons are next to their nuclei), and therefore we resort to strongly concatenable contextual process as in Theorem 2.

For example, let us consider again the net N in Figure 5. In $\mathcal{B}(N)$ we find the basic arrows $t'_0 \colon a \otimes c^- \to c_-$, $t''_0 \colon c^- \otimes a \to c_-$, $t'_1 \colon b \otimes c^- \to c_-$, $t''_1 \colon c^- \otimes b \to c_-$ and $t'_2 \colon c \to \varnothing$, with $t'_0 = \gamma_{a,c^-}; t''_0$ and $t'_1 = \gamma_{b,c^-}; t''_1$. The concurrent execution of t_0 and t_1 with shared context can be written as $(id_{c^+} \otimes \gamma_{c^-,a} \otimes id_b); (id_{c^2} \otimes t''_0 \otimes t''_1)$. This time it differs from $t'_0 \otimes id_{c^2} \otimes t''_1$ which, having source and target not in $S_N^{\otimes}$, does not correspond to any process. The concurrent execution of t_0 and t_1 with different contexts can be instead written as $\alpha = id_{c^+} \otimes t''_0 \otimes id_{c^+} \otimes t''_1$, and the terms $\alpha; (id_c \otimes t'_2)$ and $\alpha; (t'_2 \otimes id_c)$ denote different processes: in the former t_1 causes t_2 and in the latter t_0 causes t_2.

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 [6] 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.

Concluding Remarks and Future Work

Building on a illuminating suggestion of Meseguer in [9], we have shown a way to extend the algebraic semantics of PT nets proposed in [10] 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. Furthermore, in treating the individual token philosophy, we have renounced to the symmetry of the monoidal category, being then able to select only the symmetries consistent with our computational interpretation in terms of strongly concatenable contextual processes.

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 [17,18].

Acknowledgements. Thanks to José Meseguer and the referees for helpful suggestions.

References

1. P. Baldan, A. Corradini, and U. Montanari. An event structure semantics for P/T contextual nets: Asymmetric event structures. In *Proc. FoSSaCS'98*, vol. 1378 of *Lect. Notes in Comput. Sci.*, pp. 63–80. Springer, 1998.
2. E. Best and R. Devillers. Sequential and concurrent behaviour in Petri net theory. *Theoretical Computer Science*, 55:87–136, 1987.
3. R. Bruni, J. Meseguer, U. Montanari, and V. Sassone. A comparison of Petri net semantics under the collective token philosophy. In *Proc. ASIAN'98*, vol. 1538 of *Lect. Notes in Comput. Sci.*, pp. 225–244. Springer, 1998.
4. S. Christensen and N.D. Hansen. Coloured Petri nets extended with place capacities, test arcs and inhibitor arcs. In *Applications and Theory of Petri Nets*, vol. 691 of *Lect. Notes in Comput. Sci.*, pp. 186–205. Springer, 1993.
5. P. Degano, J. Meseguer, and U. Montanari. Axiomatizing the algebra of net computations and processes. *Acta Inform.*, 33(7):641–667, 1996.
6. F. Gadducci and U. Montanari. Axioms for contextual net processes. In *Proc. ICALP'98*, vol. 1443 of *Lect. Notes in Comput. Sci.*, pp. 296–308. Springer, 1998.
7. U. Goltz and W. Reisig. The non-sequential behaviour of Petri nets. *Inform. and Comput.*, 57:125–147, 1983.
8. R. Janicki and M. Koutny. Semantics of inhibitor nets. *Inform. and Comput.*, 123:1–16, 1995.
9. J. Meseguer. Rewriting logic as a semantic framework for concurrency: A progress report. In *Proc. CONCUR'96*, vol. 1119 of *LNCS*, pp. 331–372. Springer, 1996.
10. J. Meseguer and U. Montanari. Petri nets are monoids. *Inform. and Comput.*, 88(2):105–155, 1990.
11. J. Meseguer, U. Montanari, and V. Sassone. On the semantics of place/transition Petri nets. *Mathematical Structures in Computer Science*, 7:359–397, 1997.
12. 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.
13. U. Montanari and F. Rossi. Contextual nets. *Acta Inform.*, 32:545–596, 1995.
14. C.A. Petri. *Kommunikation mit Automaten.* Ph.D. thesis, Institut für Instrumentelle Mathematik, Bonn, 1962.
15. W. Reisig. *Petri Nets: An Introduction.* EACTS Monographs on Theoretical Computer Science. Springer, 1985.
16. G. Ristori. *Modelling Systems with Shared Resources via Petri Nets.* Ph.D. thesis, Dipartimento di Informatica, Università di Pisa, 1994.
17. V. Sassone. An axiomatization of the algebra of Petri net concatenable processes. *Theoretical Computer Science*, 170:277–296, 1996.
18. V. Sassone. An axiomatization of the category of Petri net computations. *Mathematical Structures in Computer Science*, 8:117–151, 1998.
19. R.J. van Glabbeek and G.D. Plotkin. Configuration structures. In *Proc. LICS'95*, pp. 199–209. IEEE Press, 1995.
20. W. Vogler. Efficiency of asynchronous systems and read arcs in Petri nets. In *Proc. ICALP'97*, vol. 1256 of *Lect. Notes in Comput. Sci.*, pp. 538–548. Springer, 1997.
21. W. Vogler. Partial order semantics and read arcs. In *Proc. MFCS'97*, vol. 1295 of *Lect. Notes in Comput. Sci.*, pp. 508–517. Springer, 1997.
22. 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.