Zero-Safe Nets: The Individual Token Approach*

Roberto Bruni and Ugo Montanari

Dipartimento di Informatica, Università di Pisa, Italia.
bruni,ugo@di.unipi.it.

Abstract. In this paper we provide both an operational and an abstract concurrent semantics for zero-safe nets under the *individual token* philosophy. The main feature of zero-safe nets is a primitive notion of *transition synchronization*. Besides ordinary places, called *stable* places, zero-safe nets come equipped with *zero* places, which are empty in any stable marking. *Connected transactions* represent *basic atomic computations* of the system between stable markings. They must satisfy two main requirements: 1) to model interacting activities which cannot be decomposed into disjoint sub-activities, and 2) not to consume stable tokens which were generated in the same transaction. Zero tokens acts as triggers for the firings of the transitions which compose the transaction. The abstract counterpart of a zero-safe net consists of a P/T net where each transition locates a distinguished transaction. In the second part of the paper, following the *Petri nets are monoids* approach, we make use of category theory to analyze and motivate our framework. More precisely, the operational semantics of zero-safe nets is characterized as an adjunction, and the derivation of abstract P/T nets as a coreflection.

1 Introduction

Petri nets [18] are one of the most attractive models of concurrency, which also offers a basic concurrent framework often used as a semantic foundation on which to interpret many concurrent languages [21, 10, 17, 6, 8, 1]. However, the basic net model does not have any synchronization mechanism among transitions, while this feature is essential to write modular and expressive programs. For

* Research supported by Office of Naval Research Contracts N00014-95-C-0225 and N00014-96-C-0114, National Science Foundation Grant CCR-9633363, and by the Information Technology Promotion Agency, Japan, as part of the Industrial Science and Technology Frontier Program "New Models for Software Architechture" sponsored by NEDO (New Energy and Industrial Technology Development Organization). Also research supported in part by U.S. Army contract DABT63-96-C-0096 (DARPA); CNR Integrated Project *Metodi e Strumenti per la Progettazione e la Verifica di Sistemi Eterogenei Connessi mediante Reti di Comunicazione*; and Esprit Working Groups *CONFER2* and *COORDINA*. Research carried on in part while the second author was on leave at Computer Science Laboratory, SRI International, Menlo Park, USA, and visiting scholar at Stanford University.

instance, all the above translations involve complex constructions for the net defining the synchronized composition of two programs.

Zero-safe nets (also *ZS nets*), introduced in [4], extend Petri nets along this direction, coming equipped with a very general notion of transition synchronization as a built-in feature. ZS nets are based on the notion of *zero* places. Tokens produced in a zero place act as triggers for the firing of transitions which are able to consume them. A distinguished set of *stable* places is also present. Stable markings (consisting only of stable tokens) describe the abstract-level markings, whilst non-stable markings (those involving zero tokens) define non-observable states of the refined model. A synchronized evolution of a ZS net starts at some stable marking, evolves through non-observable states and finally leads to a new observable state. A 'refined' ZS net and an 'abstract' Petri net are supposed to model the same given system. The latter offers the synchronized view and the former specifies how every transition of the latter is actually achieved as a different coordinated collection of firings, called *transactions*. However, the concurrent semantics of an operational model is usually defined by considering as equivalent all the computations where the same concurrent events are executed in different orders. Thus, we would like to identify those transactions which are equivalent from a *concurrent* viewpoint. The simplest approach, presented in [4], relies on the *collective token* philosophy [9], CTph for short. This school of thought identifies all the firing sequences obtained by repeatedly permuting pairs of concurrently enabled firings. The major drawback of this approach is that causal dependencies on zero tokens are lost. It follows that the class of computations captured by abstract nets may turn out to be far too generic for many applications. In this paper we present an alternative approach based on the *individual token* philosophy [9], ITph for short, where a wider class of aspects can be taken into account. In fact, we identify transactions which refer to isomorphic Goltz-Reisig processes [11]. The induced equivalence classes are called *connected transactions*.

The ZS net MS in Fig. 1 will be our running example. Net MS represents a *multicasting system*. As in a broadcasting system, an agent can simultaneously send the same message to an unlimited number of receivers, but here the receivers are not necessarily all the remaining agents, and thus several one-to-many communications can take place concurrently. Each token in place a is a different *active* (i.e., ready to communicate) agent. Transition *new* permits to create an unlimited number of agents. A firing of *send* opens a one-to-many communication: a message is put in the buffer z and the agent is suspended until the end of the transaction. A firing of *copy* adds a new copy of the message. A firing of *receive* synchronizes an active agent with a copy of the message and then suspends the agent. The transaction is completed when all the copies have been received. At the end of a session, all the suspended agents are moved into place b. Transition *reset* makes an agent active again. We call *copy policy* any strategy for making copies of the messages in the buffer. E.g., in the *sequential copying* policy, every time *copy* produces two copies of the message in the buffer, at most one of them is used to produce other copies. In the CTph, only transmis-

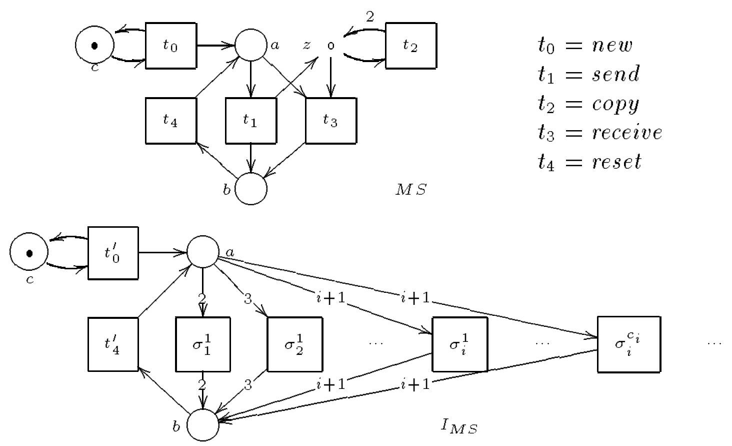


Fig. 1. A ZS net for a multicasting system, and its causal abstract net (we extend the "circles and boxes" representation for nets by drawing smaller circles for zero places).

sions differing for the number of involved agents can be distinguished. E.g., the sequential copy $sc_4 = \mathit{send}\text{-}\mathit{copy}\text{-}\{\mathit{receive}, \mathit{copy}\}\text{-}\{\mathit{receive}, \mathit{copy}\}\text{-}\mathit{2receive}$ and the balanced copy $bc_4 = \mathit{send}\text{-}\mathit{copy}\text{-}\mathit{2copy}\text{-}\mathit{4receive}$ are equivalent one-to-four transmissions [4]. On the contrary, the ITph distinguishes different copy policies. The resulting infinite *causal abstract* P/T net I_{MS} is displayed in Fig. 1. Net I_{MS} comes equipped with a *causal refinement morphism* ϵ_{MS} to the ZS net MS. Morphism ϵ_{MS} maps the places of I_{MS} into the homonymous stable places of MS, and each transition of I_{MS} into a *connected transaction* of MS (i.e., transition σ_n^k corresponds to the one-to-n transmission which follows the k-th codified copy policy). For each one-to-n transmission there are c_n copy policies (e.g., $c_4 = 2$, and the transitions σ_4^1 and σ_4^2 identify the equivalence classes of sc_4 and bc_4).

The paper is organized as follows: after briefly recalling some basic definitions of net theory, in Section 3 we present ZS nets and their operational and abstract ITph semantics. We introduce the notion of *causal firing sequences*, as enriched firing sequences. This allows a concise representation of concatenable processes, and has a suggestive implementation on a machine whose states are collections of token-stacks. Then, the evolution of a ZS net is defined in terms of equivalence classes of causal firing sequences (satisfying some additional requirement), and ordinary P/T nets are defined as the abstract counterparts of ZS nets. In Section 4, by employing some elementary category theory, we give evidence that our constructions are *natural*. Both the operational semantics of ZS nets and the derivation of the abstract P/T nets are characterized as two universal constructions (the former is an *adjunction* and the latter is a *coreflection*), following the *Petri nets are monoids* style. The universal properties of the two constructions state that they are the 'best' possible choices. Due to space limitation, some proofs are omitted and others just sketched.

2 Preliminaries

A *net* N is a triple $(S_N, T_N; F_N)$, where $S_N \neq \emptyset$ is the set of *places* $a, a', \ldots$, T_N is the set of *transitions* $t, t', \ldots$ (with $S_N \cap T_N = \emptyset$), and $F_N \subseteq (S_N \times T_N) \cup (T_N \times S_N)$ is called the *flow relation*. We will denote $S_N \cup T_N$ by N whenever no confusion arises. Subscripts will be omitted if they are obvious from the context. For $x \in N$, the set ${}^\bullet x = \{y \in N \mid yFx\}$ ($x^\bullet = \{y \in N \mid xFy\}$) is called the *pre-set* (*post-set*) of x. Let also ${}^\circ N = \{x \in N \mid {}^\bullet x = \emptyset\}$ and $N^\circ = \{x \in N \mid x^\bullet = \emptyset\}$ be the sets of *initial* and *final elements* of N, resp. A place a is said to be *isolated* iff ${}^\bullet a \cup a^\bullet = \emptyset$. We assume that for any transition t, ${}^\bullet t \neq \emptyset$.

A *P/T net* is a tuple $N = (S, T; F, W, u_{\mathrm{in}})$ s.t. $(S, T; F)$ is a net, function $W : F \longrightarrow \mathbb{N}$ assigns a positive *weight* to each arc and multiset $u_{\mathrm{in}} : S \longrightarrow \mathbb{N}$ is the *initial marking*. Relation F may be seen as a function $F : ((S \times T) \cup (T \times S)) \longrightarrow \{0, 1\}$, with $xFy \iff F(x, y) \neq 0$. Then, if we replace $\{0, 1\}$ with $\mathbb{N}$, F becomes a *multiset* relation and W is unnecessary.

A *marking* $u : S \longrightarrow \mathbb{N}$ is a finite multiset of places. It can be written either as $u = \{n_1 a_1, \ldots, n_k a_k\}$ where $n_i \in \mathbb{N}$, $n_i > 0$ (if $n_i = 0$ then the corresponding term $n_i a_i$ is safely omitted) dictates the number of occurrences (*tokens*) of the place a_i in u, i.e. $n_i = u(a_i)$, or as a formal sum $u = \bigoplus_{a_i \in S} n_i a_i$ (the order of summands is immaterial, and the addition is defined by taking $(\bigoplus_i n_i a_i) \oplus (\bigoplus_i m_i a_i) = (\bigoplus_i (n_i + m_i) a_i)$ and 0 as the neutral element). For any transition $t \in T$ let $pre(t)$ and $post(t)$ be the multisets over S such that $pre(t)(a) = F(a, t)$ and $post(t)(a) = F(t, a)$ $\forall a \in S$.

The interleaving behaviour of a net is usually described in terms of firing sequences. Given a P/T net N let u and u' be two markings of N. Then, a transition $t \in T_N$ is *enabled at* u iff $pre(t)(a) \leq u(a)$, $\forall a \in S_N$. Moreover, we say that u evolves to u' under the *firing* of t, written $u[t\rangle u'$, if and only if t is enabled at u and $u'(a) = u(a) - pre(t)(a) + post(t)(a)$, $\forall a \in S$. A *firing sequence* from u_0 to u_n is a sequence of markings and firings such that $u_0[t_1\rangle u_1 \ldots u_{n-1}[t_n\rangle u_n$. Given a marking u of N the set $[u\rangle$ of its *reachable markings* is the smallest set of markings such that $u \in [u\rangle$, and moreover $\forall u' \in [u\rangle$ such that $u'[t\rangle u''$ for some transition t, then $u'' \in [u\rangle$. Besides firings and firing sequences, *steps* and *steps sequences* are introduced. A step allows the simultaneous execution of several independent transitions. Eventually, we say that a net is *safe* if, for all reachable markings, a bound n can be given for the number of tokens in each place, i.e. $\forall u \in [u_{\mathrm{in}}\rangle, \forall a \in S,\ u(a) \leq n$.

3 Zero Safe Computations

We augment P/T nets with special places called zero places. Their role is to coordinate the atomic execution of complex collections of transitions.

Definition 1 (ZS net). A *zero-safe* net (*ZS net* for short) is a 5-tuple $B = (S_B, T_B; F_B, u_B; Z_B)$ where $N_B = (S_B, T_B; F_B, u_B)$ is the *underlying* P/T net, and the set $Z_B \subseteq S_B$ is the set of *zero places*. The places in $S_B \setminus Z_B$ are called *stable places*. A *stable marking* is a multiset of stable places.

Stable markings describe *observable* states of the system. The presence of some zero tokens in a marking makes it unobservable (e.g., *non-stable*). State changes are given in terms of *connected steps*. A connected step may involve the synchronization of several transitions, but it can be applied only if the starting state contains enough stable tokens to enable all the transitions independently. No token can be left on zero places at the end of the step (neither can be found there at the beginning of the step). Thus, all the zero tokens which are produced are also consumed in the same step. *Connected transactions* are *atomic* connected steps which consume all the stable tokens of the starting state.

In the ITph, a marking may be seen as an indexed (over the places of the net) collection of ordered sequences of tokens, and the firing of a transition specify which tokens (of each ordered sequence) are consumed and also the correspondence between each token in the reached marking with either a produced token or an idle token of the original marking. Using multisets instead of ordered sequences would make it impossible to recognize which token was produced by which firing, as it happens for the CTph.

Example 1. In our running example, suppose that the current marking is $\{a, b\}$. If t_4 fires, then a new token is produced in place a. Then, a firing of t_1 consumes a token from place a. In the ITph approach, it makes a difference if t_1 gets the token produced by t_4 or the one already present in a (in the former case the firing of t_1 causally depends on that of t_4 while in the latter case the firings of t_1 and of t_4 are concurrent activities). In the CTph approach the two firings are always concurrent, since the initial marking enables both t_1 and t_4, i.e., the execution of t_4 does not modify the enabling condition of t_1.

The Stacks Based Approach. The approach we propose is very similar to the one adopted in [19]: we choose a canonical interpretation of the tokens that are to be consumed and produced in a firing and we introduce *permutation firings* with the task of rearranging the orderings of the indexed sequences of tokens. A marking becomes a collection of stacks, one for each place, that can be accessed by transitions through a firing to extract and to insert tokens. A permutation firing is just a re-organization of the current state (i.e., of the stacks). We will denote the token stack associated to a certain place a with the term a-stack.

Definition 2 (Causal firing, permutation firing). Let N be a P/T net, and $s = u[t\rangle u'$ be a firing of N for some marking u and transition t. We interpret firing s as a *causal firing* by assuming that s consumes the 'first' $pre(t)(a)$ tokens of the a-stack of u and produces the 'first' $post(t)(a)$ tokens of the a-stack of u', for each place a. Given a marking $u = \{n_a a\}_{a \in S_N}$ of N, a *symmetry* p on u is a vector of *permutations* $p = \langle \pi_a \rangle_{a \in S_N}$ with $\pi_a \in \Pi(n_a)$, $\forall a \in S_N$, i.e. each π_a is a permutation of n_a elements. We denote by $\Pi(u)$ the set of all symmetries on u. Each symmetry p on u induces a *permutation firing* $s = u[p\rangle u$ on the net. A *causal firing sequence* is a finite sequence $\omega = s_1 \cdots s_n$ of causal and permutation firings such that $s_i = u_{i-1}[X_i\rangle u_i$ with $X_i \in T_N \cup \Pi(u_{i-1})$ for $i = 1, \ldots, n$. We say that ω *starts at* u_0 (written $O(\omega) = u_0$) and *ends in* u_n (written $D(\omega) = u_n$).

Example 2. Let N_{MS} be the underlying net of the ZS net MS in Fig. 1. A causal firing sequence for N_{MS} is $\omega = \{b,c\}[t_0\rangle\{a,b,c\}\ [t_4\rangle\{2a,c\}\ [t_1\rangle\{a,b,c,z\}$ $[t_3\rangle\{2b,c\}$. At the beginning the stacks of places a and z are empty and the stacks of places b and c contain one token each. After the firing of t_0 the token in the c-stack is replaced by a new one and a token is also inserted in the a-stack. The firing of t_4 consumes the token in the b-stack and puts a new token on top of the a-stack. Transition t_1 consumes the token on top of the a-stack and inserts a token both in the b-stack and in the z-stack. The firing of t_3 consumes the unique token in the z-stack and also the token produced by t_0 in the a-stack, and it inserts a token on top of the b-stack. Since the sequence ω does not involve any symmetry, it follows that the latest tokens produced are the first to be consumed next. To represent the sequence where t_1 depends on t_0 (and t_3 depends on t_4) we have two possibilities. The first one is to execute t_0 after t_4 (they are concurrently enabled), thus obtaining the sequence $\omega' = \{b,c\}[t_4\rangle\{a,c\}[t_0\rangle\{2a,c\}[t_1\rangle\{a,b,c,z\}[t_3\rangle\{2b,c\}$. The second possibility is to reorganize the a-stack just before the execution of t_1. This can be done via a symmetry $p = \langle(1\ 2)_a\rangle \in \Pi(2a \oplus c)$, thus obtaining the sequence $\omega'' = \{b,c\}[t_0\rangle\{a,b,c\}[t_4\rangle\{2a,c\}[p\rangle\{2a,c\}[t_1\rangle\{a,b,c,z\}[t_3\rangle\{2b,c\}$.

Review of Concatenable Processes. Causal firing sequences define a correspondence among the tokens produced and consumed via firings. This is due to the implicit orders which are imposed on the markings and is strictly related to a *process* view of computations. *Concatenable processes* [5, 20] are obtained from processes by imposing a total ordering on the origins that are instances of the same place and, similarly, on the destinations.

A net K is a *deterministic occurrence net* iff $\forall a \in S_K,\ |{}^\bullet a| \leq 1 \wedge |a^\bullet| \leq 1$ and F_K^* is acyclic (F^* denotes the reflexive and transitive closure of relation F), i.e., $\forall x, y \in K,\ xF_K^*y \wedge yF_K^*x \Longrightarrow x = y$). A (Goltz-Reisig) *process* for a P/T net N is a mapping $P : K \longrightarrow N$ from an occurrence net K to N such that $P(S_K) \subseteq S_N$, $P(T_K) \subseteq T_N$, ${}^\circ K \subseteq S_K$, and $\forall t \in T_K,\ \forall a \in S_N,\ F_N(a, P(t)) = \left|P^{-1}(a) \cap {}^\bullet t\right| \ \wedge\ F_N(P(t), a) = \left|P^{-1}(a) \cap t^\bullet\right|$. As usual we denote the set of *origins* (i.e., minimal or initial places) and *destinations* (i.e., final or maximal places) with $O(K) = {}^\circ K$ and $D(K) = K^\circ \cap S_K$, resp. Two processes P and P' of N are *isomorphic* and thus identified if there exists an isomorphism $\psi : K_P \longrightarrow K_{P'}$ such that $P' \circ \psi = P$.

Given a set S with a labelling function $l : S \longrightarrow S'$, a *label-indexed ordering function* for l is a family $\beta = \{\beta_a\}_{a \in S'}$ of bijections, where $\beta_a : l^{-1}(a) \longrightarrow \{1, \ldots, \left|l^{-1}(a)\right|\}$. A *concatenable process* for a P/T net N is a triple $C = (P,\ {}^\circ\ell, \ell^\circ)$ where $P : K \longrightarrow N$ is a process for N and ${}^\circ\ell,\ \ell^\circ$ are label-indexed ordering functions for the labelling function P restricted to $O(K)$ and $D(K)$, resp. Two concatenable processes C and C' are isomorphic if P_C and $P_{C'}$ are isomorphic via a mapping preserving all the orderings.

A partial binary operation $\_;\_$ (associative up to iso and with identities) of concatenation of concatenable processes (whence their names) can be easily defined: we take as source (target) the image through P of the initial (maximal) places of K_P; then the composition of $C = (P,\ {}^\circ\ell, \ell^\circ)$ and $C' = (P',\ {}^\circ\ell', \ell'^\circ)$

is realized by merging, when it is possible, the maximal places of K_P with the initial places of $K_{P'}$ according to their labelling and ordering functions so to match those places one-to-one. Concatenable processes admit also a monoidal *parallel* composition $\_ \otimes \_$, which can be represented by putting two processes side by side. Due to space limitation, we refer the interested reader to [5] for the formal definitions.

3.1 From Causal Sequences to Processes.

It may be easily noticed that each causal firing sequence uniquely determines a concatenable process. Informally the construction associates an elementary (concatenable) process to each causal and permutation firing.

From causal firings to processes. Let N be a P/T net and $s = u[t\rangle u'$ be a causal firing, with $u = \{n_a a\}_{a \in S_N}$, $pre(t) = \{h_a a\}_{a \in S_N}$, $post(t) = \{k_a a\}_{a \in S_N}$. The associated concatenable process is $pr(s) = (P : K \longrightarrow N, {}^\circ\ell, \ell^\circ)$, where:

- $T_K = \{\tilde{t}\}$, $P(\tilde{t}) = t$; $S_K = \{\tilde{a}_i \mid a \in S_N,\ 1 \le i \le n_a + k_a\}$, $P(\tilde{a}_i) = a$;
- ${}^\bullet\tilde{t} = \{\tilde{a}_i \mid a \in S_N,\ 1 \le i \le h_a\}$, $\tilde{t}^\bullet = \{\tilde{a}_i \mid a \in S_N,\ h_a + 1 \le i \le h_a + k_a\}$, thus $O(K) = \{\tilde{a}_i \in S_K \mid i \le h_a \vee i \ge h_a + k_a + 1\}$ and $D(K) = \{\tilde{a}_i \in S_K \mid i \ge h_a + 1\}$;
- $\forall \tilde{a}_i \in O(K)$, ${}^\circ\ell_a(\tilde{a}_i) = \begin{cases} i & \text{if } 1 \le i \le h_a \\ i - k_a & \text{if } h_a + 1 + k_a \le i \le n_a + k_a \end{cases}$;
- $\forall \tilde{a}_i \in D(K)$, $\ell^\circ_a(\tilde{a}_i) = i - h_a$.

A brief explanation is necessary. The occurence net K contains a unique transition $\tilde{t}$ (mapped onto transition t of N), and a place for each consumed, produced, and idle token of s. For each place $a \in S_N$ we need exactly $n_a + k_a$ different places in S_K. We denote the generic i-th place associated to place a with $\tilde{a}_i$. The set $\{\tilde{a}_i \mid a \in S_N,\ 1 \le i \le h_a\}$ represents the tokens which are consumed by the causal firing of t, i.e. the 'first' h_a tokens of each a-stack in the starting state. The set $\{\tilde{a}_i \mid a \in S_N,\ h_a + 1 \le i \le h_a + k_a\}$ represents the tokens which are produced by the causal firing of t, i.e. the 'first' k_a tokens of each a-stack in the ending state. The set $\{\tilde{a}_i \mid a \in S_N,\ h_a + k_a + 1 \le i \le n_a + k_a\}$ contains the remaining idle tokens, i.e. the 'last' $n_a - h_a$ token of each a-stack of both u and u'. Functions ${}^\circ\ell$ and ℓ° are defined accordingly with this assumptions.

From permutation firings to processes. Let N be a P/T net and $s = u[p\rangle u'$ be a permutation firing, with $u = \{n_a a\}_{a \in S_N}$ and $p = \langle \pi_a \rangle_{a \in S_N}$. The associated concatenable process $pr(s) = (P : K \longrightarrow N, {}^\circ\ell, \ell^\circ)$ is defined as follows:

- $T_K = \emptyset$ (it follows that $O(K) = D(K) = S_K$);
- $S_K = \{\tilde{a}_i \mid a \in S_N,\ 1 \le i \le n_a\}$, $P(\tilde{a}_i) = a$;
- $\forall \tilde{a}_i \in O(K)$, ${}^\circ\ell_a(\tilde{a}_i) = i$; $\forall \tilde{a}_i \in D(K)$, $\ell^\circ_a(\tilde{a}_i) = \pi_a(i)$.

In this case the set of transitions is empty and all the tokens stay idle. The generic place $\tilde{a}_i$ of k denotes the instance of place a which corresponds to the i-th token (from the top) of the a-stack of the starting state. The re-organization induced by the permutation firing is provided by the functions ${}^\circ\ell$ and ℓ°.

The concatenable process associated to a (finite) causal firing sequence is given by the concatenation of the concatenable processes associated to each step of the given sequence. In what follows we denote with $pr(\omega)$ the concatenable process associated with the causal firing sequence ω (up to iso).

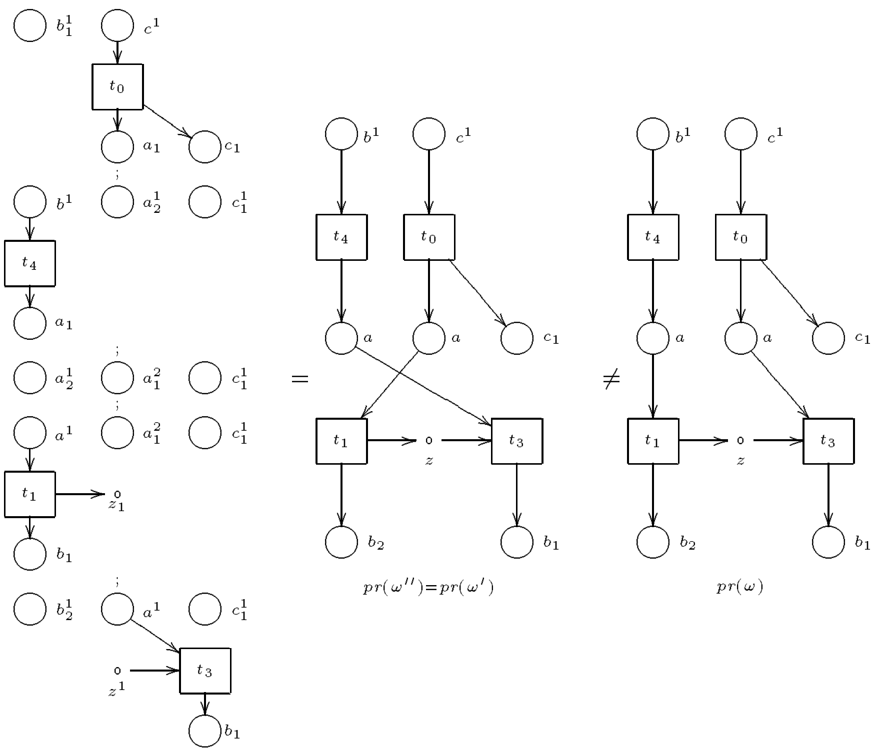


Fig. 2. The concatenable processes derived from sequences ω'', ω', and ω of Ex. 2.

Example 3. The concatenable processes derived from the sequences of Ex. 2 are presented in Fig. 2 (we use the standard notation that labels the places and transitions of the occurence net K with their images in N; a superscript for any initial place and a subscript for any final place denotes the value of ${}^{\circ}\ell$ and ℓ°, resp.), the construction of $pr(\omega'')$ being explained in details.

Terminology. Let us introduce some properties of processes that we will use extensively. A process is *active* iff it includes at least one transition, *inactive* otherwise. An active process is *decomposable into parallel activities* iff it is the parallel composition of two (or more) active processes. If such a decomposition does not exist, then the process is called *connected*. A connected process may involve idle places, but it does not admit disjoint activities. The resources which are first produced and then consumed (i.e., the 'inner' places) are called *evolution places*. More formally, a concatenable process $C = (P : K \longrightarrow N, {}^{\circ}\ell, \ell^{\circ})$ is *connected* iff the set of transitions of K is non-empty and moreover, for each pair (t, t') of transitions of K there exists an undirected path (through the arcs of the flow relation) connecting t and t'. Process C is *full* iff it does not contain idle (i.e., isolated) places (i.e., $\forall a \in S_K,\ |{}^{\bullet}a| + |a^{\bullet}| \geq 1$). Finally, the set of *evolution places* of process C is $E_C = \{P(a) \mid a \in K,\ |{}^\bullet a| = |a^\bullet| = 1\}$.

3.2 Operational Semantics

A causal firing sequence is essentially a linearization of a concatenable process, and more than one sequence[2] can correspond to the same (up to iso) concatenable process. Thus we consider the equivalence over sequences induced by isomorphic processes. Moreover, we will notice that, for the kind of sequences under consideration, the label-indexed ordering functions of origins and destinations are no longer important, so we base the equivalence on the Goltz-Reisig processes.

Definition 3 (Equivalent causal firings). Given a net N, we say that two causal firing sequences ω and ω' are *causally equivalent*, written $\omega \approx \omega'$ iff $pr(\omega) = (P,\ {}^\circ\ell, \ell^\circ)$ and $pr(\omega') = (P',\ {}^\circ\ell', \ell'^\circ)$ with process P isomorphic to P'. The equivalence class of ω is denoted with $[\![\omega]\!]_\approx$. We use ξ to range over equivalence classes. Since relation $\approx$ respects the initial and final marking, we extend the notation letting $O(\xi) = O(\omega)$ and $D(\xi) = D(\omega)$, for $\xi = [\![\omega]\!]_\approx$.

Definition 4 (Connected step and transaction). Given a ZS net B, let $\omega = s_1 \cdots s_n$ be a causal firing sequence of the P/T net N_B. The equivalence class $\xi = [\![\omega]\!]_\approx$ is a *connected step* of B, written $O(\xi)[\![\xi\rangle D(\xi)$, if (i) $O(\omega)$ and $D(\omega)$ are stable markings, and (ii) $E_{pr(\omega)} \subseteq Z_B$. A *connected step sequence* is a finite sequence $u_0[\![\xi_1\rangle u_1 \ldots u_{n-1}[\![\xi_n\rangle u_n$ and we say that u_n is reachable from u_0. Furthermore, the connected step ξ is a *connected transaction* of B if (iii) $pr(\omega)$ is connected, and (iv) $pr(\omega)$ is full. We denote with Ξ_B the set of all the connected transactions over B. Properties (i)-(iv) impose conditions only over the Goltz-Reisig process associated with $pr(\omega)$, and thus are preserved by equivalence $\approx$.

We claim that connected transactions are a good definition for the basic behaviours of the systems. Our assertion is supported by the fact that connected transactions denote *atomic computations that cannot be extended further*. Atomicity follows immediately from the connectedness of the associated processes. The second argument deserves a more precise explanation. An atomic behaviour can be extended if there exists a broader atomic behaviour of which the former is a sub-part. From our viewpoint, the only interaction allowed in a ZS net is given by the flow of tokens through zero places. Since connected steps and transactions start and also end in stable markings, it is impossible to hook them in a wider atomic computation by means of zero tokens. This is very clear for transactions, because they consume all the needed resources. This is not the case of connected steps, since it could be possible for some resource to stay idle during the whole sequence of moves. However this kind of resources are stable and not connected to the rest of the step, thus, any other activity involving them is intrinsically concurrent w.r.t. the step under consideration. It follows that any wider behaviour extending a connected step is not atomic (i.e., it can be expressed in terms of concurrent components).

[2] Sequences differing in the order in which concurrent firings are executed or for the way in which equivalent symmetries are performed are identified.

Example 4. Let us consider the ZS net MS of Fig. 1. The equivalence class of the causal firing sequence $\{a\}[t_1\rangle\{b,z\}[t_4\rangle\{a,z\}[t_3\rangle\{b\}$ is not a connected step since the prop. (ii) is not satisfied. Class $[\![\{2a,c\}[t_1\rangle\{a,b,c,z\}[t_3\rangle\{2b,c\}[t_0\rangle\{a,2b,c\}]\!]_\approx$ is a connected step but not a connected transaction since the constraint (iii) is not satisfied. The class of $\{4a\}[t_1\rangle\{3a,b,z\}[t_2\rangle\{3a,b,2z\}[t_3\rangle\{2a,2b,z\}[t_3\rangle\{a,3b\}$ is a connected step but not a connected transaction since the prop. (iv) is not satisfied. The class of $\{5a\}[t_1\rangle u_1[t_2\rangle u_2[t_2\rangle u_3[t_2\rangle u_4[t_3\rangle u_5[t_3\rangle u_6[t_3\rangle u_7[t_3\rangle\{5b\}$, where intermediate markings are the obvious ones, is a connected transaction.

3.3 Abstract Semantics

Next, we define an abstract view of the system modelled by a ZS net. Since transactions rewrite multisets of stable tokens, it is natural to choose a net as a candidate for the abstraction. Since the ordering of tokens in the pre-set (post-set) of a transition is useless we should abstract from it. This is already done via the equivalence classes of causal firing sequences. When restricted to connected steps, this equivalence intuitively corresponds to limit the symmetries of permutation firings to be vectors of permutations over the zero places only, with the assumption that the stable tokens which are produced in a transaction are not reused during the same transaction. The last statement was also the basis for the CTph approach.

Example 5. Consider the ZS net MS in Fig. 1. Let $\omega = \{2a\}[t_1\rangle\{a,b,z\}[t_3\rangle\{2b\}$, $s = \{2a\}[p\rangle\{2a\}$ and $s' = \{2b\}[p'\rangle\{2b\}$, where p and p' are the symmetries which swap the two tokens in a and b, resp. The causal sequences ω, $s\omega$, and $\omega s'$ define the same connected transaction $\xi = [\![\omega]\!]_\approx$, but $pr(s\omega) \neq pr(\omega) \neq pr(\omega s')$. If we represent the connected transaction ξ as a transition t of a net, then its preset (as well as its postset) is an unordered multiset. This means that when t fires it is impossible to distinguish among the two tokens in b that it produces, and also among the two tokens in a that it consumes. We can conclude that it makes no sense to have many different transitions to represent behaviours that we cannot reproduce at the abstract level. Thus we are forced to identify $pr(s\omega)$, $pr(\omega)$, $pr(\omega s')$, and also $pr(s\omega s')$.

Definition 5 (Causal abstract net). Let $B = (S_B, T_B; F_B, u_B; Z_B)$ a ZS net. Net $I_B = (S_B \backslash Z_B, \Xi_B; F, u_B)$, with $F(a,\delta) = pre(\delta)(a)$ and $F(\delta,a) = post(\delta)(a)$, is the *causal abstract net* of B, where Ξ_B is the set of all the connected transaction of B, and $pre(\delta)$ and $post(\delta)$ denote the multisets $O(\delta)$ and $D(\delta)$, resp.

Example 6. We conclude this section by illustrating the causal abstract net of the multicasting system. The net I_{MS} is (partially!) depicted in Fig. 1. Transition t'_0 creates a new communicating process and it corresponds to $[\![\omega_0]\!]_\approx$ with $\omega_0 = \{c\}[t_0\rangle\{a,c\}$. Similarly t'_4 is the equivalence class of the the firing of t_4 in the marking $\{b\}$. Each σ_i^k describes a different one-to-i communication, where index k identifies the copy policy. A generic one-to-i communication can be essentially described as follows: a firing of t_1 initiates the communication, then the system executes as many firings of t_2 as the number of copies of the message needed (i.e., $i-1$ since a message is already present in the buffer), and finally i firings of t_3 synchronize the messages with different active processes. In the ITph we distinguish among tokens in a same marking, which were created by different firings. In this way we have a one-to-one correspondence among copy policies and the complete[3] binary trees with exactly i leaves (we don't distinguish between 'left' and 'right' children).

For any i, the total number of copy policies can be derived as follows. For $i = 1$ there is only one tree whose root is the unique leaf. If $i = 2h + 1$ for some integer $h > 0$, it follows that one of the subtrees rooted in a child of the root is a complete binary tree with $j \leq h$ leaves, while the subtree rooted in the other child is a complete binary tree with $i - j$ leaves; for any j, we know that there are $c_j \cdot c_{i-j}$ possible trees made in this way, thus $c_i = \sum_{j=1}^{h} c_j \cdot c_{i-j}$. If $i = 2h$ for some integer $h > 0$, we adopt an analogous reasoning to deduce that for any $j < h$ there are $c_j \cdot c_{i-j}$ possible complete binary trees such that exactly j leaves belongs to one of the subtrees rooted in the children of the root. The case $j = h$ requires more attention. In fact, if the two subtrees have the same number h of leaves then there are $\frac{c_h \cdot (c_h+1)}{2}$ possible ways for choosing them. It follows that $c_i = \frac{c_h \cdot (c_h+1)}{2} + \sum_{j=1}^{h-1} c_j \cdot c_{n-j}$. Since there are no transitions in MS requiring 2 or more zero token, there are no other transactions.

4 Universal Constructions

The aim of this section is to propose an algebraic characterization of the definitions and the constructions presented in the previous section. To this purpose, we make use of some elementary concepts of category theory. The first notion consists of the *category of models* itself: objects are models and arrows represent some notion of simulation. The choice of arrows is very informative, since they complement and in a sense redefine (e.g., isomorphic objects are often identified) the meaning of models. We define a suitable category **dZPetri** (where ZS nets are considered as programs) where net morphisms satisfy an important additional condition. Then we consider a construction that exhibits an *adjunction* from **dZPetri** to a category **ZSCGraph** consisting of some kind of machines, equipped with operations and transitions between states. It is proved that this adjunction is strictly related to the semantics of ZS nets defined in the previous section. Our second construction starts from a complex category **ZSC** of ZS nets (which is however strictly related to **ZSCGraph**), having the ordinary category **Petri** of P/T nets as a subcategory, and yields a *coreflection* corresponding exactly to the construction of the causal abstract net in Def. 5.

4.1 Review of 'Petri Nets are Monoids'

Petri net theory can be profitably developed within category theory [22, 13, 2]. We follow the approach initiated in [13] (other references are [14, 5, 15, 16]).

[3] We say that a binary tree is *complete* if any internal node has exactly two children.

A (place/transition) *Petri net* is a graph $(S^{\oplus}, T, \partial_0, \partial_1)$ where the set of nodes is the free commutative monoid $S^{\oplus}$ over the set of *places* S (functions $\partial_0, \partial_1 : T \longrightarrow V$ are called *source* and *target*, resp., and we write $t : u \longrightarrow v$, with obvious meaning, to shorten the notation). A *Petri net morphism* is a graph morphism $h = (f : T \longrightarrow T', g : S^{\oplus} \longrightarrow S'^{\oplus})$ (i.e., $g(\partial_i(u)) = \partial'_i(f(u))$ for $i = 0, 1$) where g is a monoid homomorphism (this defines the category **Petri**).

In [14, 5] it has been shown that it is possible to enrich the algebraic structure of transitions in order to capture some basic constructions on nets. As an example, the forgetful functor from **CMonRPetri** [14] to **Petri** has a left adjoint which associates to each Petri net N its *marking graph* $\mathcal{C}[N]$, which corresponds to the ordinary operational semantics of N (i.e., its arrows are the step sequences of N). The objects of **CMonRPetri** are *reflexive Petri commutative monoids* (i.e., Petri nets together with a function $id : S^{\oplus} \longrightarrow T$, where T is a commutative monoid $(T, \otimes, 0)$ and ∂_0, ∂_1 and id are monoid homomorphisms), and its arrows are Petri net morphisms preserving identities and the monoidal structures.

The algebraic structure of process is well captured in [20]. There it is shown how to associate a free symmetric strict monoidal category (see Appendix A) $\mathcal{F}[N]$ to each net N in such a way that, under two suitable axioms, it characterizes the concatenable processes of N. This is due to the existence of a left adjoint functor $\mathcal{F} : \mathbf{Petri} \longrightarrow \mathbf{SSMC}^{\oplus}$ to the forgetful functor $\mathcal{U} : \mathbf{SSMC}^{\oplus} \longrightarrow \mathbf{Petri}$. Given a net N the category $\mathcal{F}[N]$ has the elements of $S_N^{\oplus}$ as objects, while its arrows are generated by the following inference rules

$$\frac{u \in S_N^{\oplus}}{id_u : u \longrightarrow u \in \mathcal{F}[N]} \qquad \frac{t : u \longrightarrow v \in T_N}{t : u \longrightarrow v \in \mathcal{F}[N]} \qquad \frac{a, b \in S_N}{c_{a,b} : a \oplus b \longrightarrow b \oplus a \in \mathcal{F}[N]}$$

$$\frac{\alpha : u \longrightarrow v,\ \beta : u' \longrightarrow v' \in \mathcal{F}[N]}{\alpha \otimes \beta : u \oplus u' \longrightarrow v \oplus v' \in \mathcal{F}[N]} \qquad \frac{\alpha : u \longrightarrow v,\ \beta : v \longrightarrow w \in \mathcal{F}[N]}{\alpha ; \beta : u \longrightarrow w \in \mathcal{F}[N]}$$

modulo the axioms expressing that $\mathcal{F}[N]$ is a strict monoidal category, and the axioms stating that the collection[4] $\{c_{u,v}\}_{u,v \in S_N^{\oplus}}$ plays the role of the symmetry natural isomorphism which makes $\mathcal{F}[N]$ into a ssmc. This axiomatization will be useful to shorten the notation in the sketched proof of Th. 16.

Theorem 6. *Given a net N, the concatenable processes of N are isomorphic to the arrows of the category $\mathcal{P}[N]$, which is the monoidal quotient of the free ssmc on N ($\mathcal{F}[N]$) modulo the axioms*

$$c_{a,b} = id_{a \oplus b} \quad \textit{if } a \neq b \in S_N, \textit{ and} \tag{1}$$

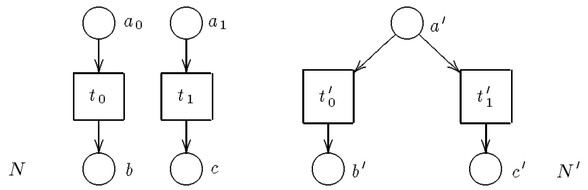
$$s; t; s' = t \quad \textit{if } t \in T_N, \textit{ and } s, s' \textit{ are symmetries.} \tag{2}$$

The previous construction provides an algebraic view of net computations which is strictly related to a process understanding of the causal behaviour of a net, but is not functorial. The main problem is that there exist reasonable morphisms of nets which cannot be extended to a monoidal functor. We illustrate

[4] Symmetries $c_{u,v}$ for $u, v \in S_N^{\oplus}$ denote any term obtained from $c_{a,b}$ for $a, b \in S_N$ by applying recursive rules analogous to axioms (3) given in Th. 11.

below the example presented in [16]. We will show that this kind of morphisms can be avoided in the category of ZS nets, our choice being justified by the necessity to preserve atomic behaviours through morphisms.

Example 7. Consider the nets N and N' pictured below and the net morphism $f : N \longrightarrow N'$ s.t. $f(t_i) = t'_i$, $f(a_i) = a'$, $f(b) = b'$ and $f(c) = c'$ for $i = 0, 1$.



Morphism f cannot be extended to a functor $\mathcal{P}[f] : \mathcal{P}[N] \longrightarrow \mathcal{P}[N']$. In fact, supposing that such an extension F exists, then $F(t_0 \otimes t_1) = F(t_0) \otimes F(t_1) = t'_0 \otimes t'_1$ by the monoidality of F. Since $t_0 \otimes t_1 = t_1 \otimes t_0$ in $\mathcal{P}[N]$, then $t'_0 \otimes t'_1 = t'_1 \otimes t'_0$ which is impossible, as the two expressions denote different processes in $\mathcal{P}[N']$.

4.2 Operational Semantics as Adjunction

Definition 7 (Category dZPetri). A *ZS net* is a Petri net where the set of places $S = L \cup Z$ is partitioned into *stable* and *zero* places. A *ZS net morphism* is a Petri net morphism $(f, g) : N \longrightarrow N'$ where homomorphism g preserves partitioning of places (i.e., if $a \in Z$ then $g(a) \in Z'^{\oplus}$ and if $a \in S \setminus Z$ then $g(a) \in (S' \setminus Z')^{\oplus}$) and satisfies the additional condition of mapping zero places into pairwise disjoint (non-empty) zero markings (*disjoint image* property). We call *disjoint* any morphism of this kind. This defines the category **dZPetri**.

Since $S^{\oplus}$ is a free commutative monoid we may represent the set of nodes of a ZS net as $L^{\oplus} \times Z^{\oplus}$, and ZS net morphisms as triples (f, g_L, g_Z), where g_L and g_Z are monoid homomorphisms on the monoids of stable and zero places, resp.

Example 8. The graph corresponding to the ZS net MS defined in Fig. 1 has the following set of arcs: $T_{MS} = \{t_0 : (c, 0) \longrightarrow (a \oplus c, 0), t_1 : (a, 0) \longrightarrow (b, z), t_2 : (0, z) \longrightarrow (0, 2z), t_3 : (a, z) \longrightarrow (b, 0), t_4 : (b, 0) \longrightarrow (a, 0)\}$.

Disjoint morphisms play a very important role here. If we restrict to consider disjoint morphisms only, then we avoid the awkward situation arising from Ex. 7. Moreover, if we identified two different zero places via a (non-disjoint) morphism then the behaviour of the abstract model might dramatically change. Since we use zero places to specify a synchronization mechanism, it is important to ensure that this mechanism is always preserved.

The next definition introduces a category of more structured models, which is reminiscent of the constructions both of marking graphs and free ssmc's.

Definition 8 (Category ZSCGraph). A *ZS causal graph* $E = ((L \cup Z)^{\oplus}, (T, \otimes, 0, id, *), \partial_0, \partial_1)$ is both a ZS net and a reflexive Petri monoid. In addition, it comes equipped with a partial function $\_ * \_$ called *horizontal composition*:

$$\frac{\alpha : (u, x) \longrightarrow (v, y),\ \beta : (u', y) \longrightarrow (v', y')}{\alpha * \beta : (u \oplus u', x) \longrightarrow (v \oplus v', y')}$$

and a collection of *horizontal swappings* $\{e_{x,y} : (0, x \oplus y) \longrightarrow (0, y \oplus x)\}_{x,y \in Z^{\oplus}}$. Horizontal composition is associative and has identities $id_{(0,x)}$ for any $x \in Z^{\oplus}$. The monoidal operator $\_ \otimes \_$ is functorial w.r.t. horizontal composition, and the *horizontal naturality* axiom $e_{x,x'} * (\beta \otimes \alpha) = (\alpha \otimes \beta) * e_{y,y'}$ holds for any $\alpha : (u, x) \longrightarrow (v, y)$ and $\beta : (u', x') \longrightarrow (v', y')$. Moreover, the following *coherence* axioms are satisfied for any $x, y, y' \in Z^{\oplus}$: $e_{x,y} * e_{y,x} = id_{(0,x \oplus y)}$, and $e_{x,y \oplus y'} = (e_{x,y} \otimes id_{(0,y')}) * (id_{(0,y)} \otimes e_{x,y'})$. A morphism h between two ZS causal graphs E and E' is a monoidal disjoint morphism which in addition respects horizontal composition and swappings This defines the category **ZSCGraph**.

Horizontal composition is the key feature of our approach. It behaves like sequential composition on zero places and like the ordinary parallel composition on stable places. This is necessary to avoid the construction of steps which reuse stable tokens. Swappings are used to specify the causality relation among produced and consumed zero tokens.

Proposition 9. *If $\alpha : (u, 0) \longrightarrow (v, 0)$ and $\alpha' : (u', 0) \longrightarrow (v', 0)$ are two transitions of a ZS causal graph then $\alpha \otimes \alpha' = \alpha' \otimes \alpha$ and $\alpha * \alpha' = \alpha \otimes \alpha'$.*

Corollary 10. *The full subcategory of* **ZSCGraph** *whose objects are Petri nets (i.e., $Z = \emptyset$) is isomorphic to* **CMonRPetri**.

Theorem 11. *The obvious forgetful functor $\mathcal{U}$:* **ZSCGraph** $\longrightarrow$ **dZPetri** *has a left adjoint $\mathcal{CG}$:* **dZPetri** $\longrightarrow$ **ZSCGraph**, *which maps a ZS net B into the ZS causal graph $\mathcal{CG}[B]$, whose arrows are generated by the following inference rules*

$$\frac{t : (u, x) \longrightarrow (v, y) \in T_B}{t : (u, x) \longrightarrow (v, y) \in \mathcal{CG}[B]} \qquad \frac{\alpha : (u, x) \longrightarrow (v, y),\ \beta : (u', x') \longrightarrow (v', y') \in \mathcal{CG}[B]}{\alpha \otimes \beta : (u \oplus u', x \oplus x') \longrightarrow (v \oplus v', y \oplus y') \in \mathcal{CG}[B]}$$

$$\frac{(u, x) \in L_B^{\oplus} \times Z_B^{\oplus}}{id_{(u,x)} : (u, x) \longrightarrow (u, x) \in \mathcal{CG}[B]} \qquad \frac{z, z' \in Z_B}{d_{z,x} : (0, z \oplus x) \longrightarrow (0, x \oplus z) \in \mathcal{CG}[\mathcal{B}]}$$

$$\frac{\alpha : (u, x) \longrightarrow (v, y),\ \beta : (u', y) \longrightarrow (v', z) \in \mathcal{CG}[B]}{\alpha * \beta : (u \oplus u', x) \longrightarrow (v \oplus v', z) \in \mathcal{CG}[B]}$$

modulo the axioms expressing that the arrows form a (strict) monoid with unit $id_{(0,0)}$, and that horizontal composition $\_ * \_$ is associative and has identities $id_{(0,x)}$, the functoriality axiom for the tensor product, and the axioms expressing that the collection of swappings $d_{x,y}$ plays the role of the 'horizontal' natural isomorphism: $d_{x,x'} * (\beta \otimes \alpha) = (\alpha \otimes \beta) * d_{y,y'}$, and $d_{z,z'} * d_{z',z} = id_{z \oplus z'}$, for any arrows $\alpha : (u, x) \longrightarrow (v, y), \beta : (u', x') \longrightarrow (v', y') \in \mathcal{CG}[B]$, and for any $z, z' \in Z_B$, where $d_{x,y}$ for $x, y \in Z_B^{\oplus}$ denotes any term obtained from the basic symmetries by applying recursively the rules:

$$\begin{aligned} d_{0,x} &= id_{(0,x)} = d_{x,0}, \\ d_{z \oplus x,y} &= (id_{(0,z)} \otimes d_{x,y}) * (d_{z,y} \otimes id_{(0,x)}), \text{ and} \\ d_{x,y \oplus z} &= (d_{x,y} \otimes id_{(0,z)}) * (id_{(0,y)} \otimes d_{x,z}). \end{aligned} \tag{3}$$

Proof. (Sketch). It follows immediately from the definition that $\mathcal{CG}[\mathcal{B}]$ is a ZS causal graph. We need to show that it is the *free* ZS causal graph on B. Let $\eta_B : B \longrightarrow \mathcal{U}[\mathcal{CG}[\mathcal{B}]]$ the disjoint ZS net morphism which is the identity on places and the obvious injection on transitions. We show that η_B is universal, i.e., for any ZS causal graph E and for any disjoint ZS net morphism $h = (f, g_L, g_Z) : B \longrightarrow \mathcal{U}[E]$, there exists a unique ZS causal graph morphism $k : \mathcal{CG}[\mathcal{B}] \longrightarrow E$ such that $h = \eta_B ; \mathcal{U}[k]$ (in **dZPetri**). Thus, morphisms k and h must agree on the generators of $\mathcal{CG}[\mathcal{B}]$ and the extension of k to tensor and horizontal composition is uniquely determined by its definition on the generators. The proof can be completed just showing that k preserves the axioms which generate $\mathcal{CG}[B]$.

The notion of adjunction between a category with 'more structure' (**ZSCGraph** in our case), and a similar category but with 'less structure' (**dZPetri**) is useful to characterize natural constructions. In fact, the left-adjoint to the usually obvious forgetful functor which deletes the 'extra' structure is unique (up to iso) and represents the best possible way for adding this structure.

Theorem 12. *When restricted to P/T nets, functor $\mathcal{CG}$ coincides with $\mathcal{C}$.*

The previous theorem shows that the algebraic semantics of ZS nets is an extension of the ordinary semantics of P/T nets. Unfortunately, the ZS causal graph $\mathcal{CG}[B]$ is still too concrete w.r.t. the operational semantics of ZS nets. Thus, we need two more axioms (analogous to axioms (1) and (2) of Th. 6).

Definition 13. Given a ZS net B, let $\mathcal{CG}[B]/\Psi$ be the quotient of the free ZS causal graph $\mathcal{CG}[B]$ generated by B in **ZSCGraph** modulo the axioms

$$d_{z,z'} = id_{(0,z\oplus z')} \;\text{ if }\; z \neq z' \in Z_B, \;\text{ and} \tag{4}$$

$$d * t * d' = t \;\text{ if }\; t \in T_B, \;\text{ and } d, d' \text{ are swappings.} \tag{5}$$

The quotient $\mathcal{CG}[B]/\Psi$ is s.t. for any ZS causal graph morphism $k : \mathcal{CG}[B] \longrightarrow E$ respecting axioms (4) and (5) (i.e., $k(d_{z,z'}) = id_{(0,k(z)\oplus k(z'))}$, and $k(d*t*d') = k(t)$), there is a unique arrow k_Ψ such that $k = Q_\Psi ; k_\Psi$ (in **ZSCGraph**), where $Q_\Psi : \mathcal{CG}[B] \longrightarrow \mathcal{CG}[B]/\Psi$ is the obvious morphism associated to the (least) congruence generated by the imposed axiomatization.

Proposition 14. *For any disjoint morphism $h : B \longrightarrow B'$ in* **dZPetri** *there exists a unique extension $\hat{h} : \mathcal{CG}[B]/\Psi \longrightarrow \mathcal{CG}[B']/\Psi$ of h in* **ZSCGraph**.

Proof. Take $k = Q'_\Psi \circ \mathcal{CG}[h] : \mathcal{CG}[B] \longrightarrow \mathcal{CG}[B']/\Psi$. Morphism k respects axioms (4) and (5) (because h is disjoint and maps transitions to transitions), thus k_Ψ is uniquely determined. Then take $\hat{h} = k_\Psi$.

Example 9. Let MS be the ZS net of our running example whose set of arcs is defined in Ex. 8. For instance the arrow $t_1 * t_3 \in \mathcal{CG}[MS]/\Psi$ has source $(2a, 0)$ and target $(2b, 0)$. Instead, notice that the arrow $(t_1 \otimes id_{(a,0)}) * (id_{(b,0)} \otimes t_3)$ goes from $(3a \oplus b, 0)$ to $(a \oplus 3b, 0)$. As another example, the following expressions are all identified in $\mathcal{CG}[MS]/\Psi$, i.e., they all denote the same arrow: $t_1 * t_2 * (t_2 \otimes t_3) * (t_3 \otimes t_3) = t_1 * t_2 * (t_2 \otimes id_{(0,z)}) * (t_3 \otimes t_3 \otimes t_3) = t_1 * t_2 * d_{z,z} * (t_2 \otimes id_{(0,z)}) * (t_3 \otimes t_3 \otimes t_3) = t_1 * t_2 * (id_{(0,z)} \otimes t_2) * (t_3 \otimes t_3 \otimes t_3) = t_1 * t_2 * (t_3 \otimes t_2) * (t_3 \otimes t_3)$.

Definition 15 (Prime Arrow). An arrow $\alpha : (u, 0) \longrightarrow (v, 0)$ of a ZS causal graph E is *prime* iff α cannot be expressed as the monoidal composition of non-trivial arrows (i.e., $\nexists \beta, \gamma \in H, \beta \neq id_{(0,0)} \neq \gamma$ such that $\alpha = \beta \otimes \gamma$).

Example 10. In our running example, some prime arrows of $\mathcal{CG}[MS]$ are $t_0, t_1 * t_3$, and $t_1 * t_2 * (t_2 \otimes t_2) * (t_3 \otimes t_2 \otimes t_3 \otimes t_3) * (t_3 \otimes t_3)$. As a counterexample, the arrow $(t_1 \otimes t_1) * d_{z,z} * (t_2 \otimes t_3) * (t_3 \otimes t_3)$ is not prime.

Theorem 16. *Given a ZS net B, there is a one-to-one correspondence between arrows $\alpha : (u, 0) \longrightarrow (v, 0) \in \mathcal{CG}[B]/\Psi$ and the connected steps of B. Moreover, if such an arrow is prime (and is not an identity) then the corresponding connected step is a connected transaction.*

Proof. (Sketch). For any arrow β of $\mathcal{CG}[B]/\Psi$ we define inductively on the structure of β a concatenable process $C(\beta)$ of N_B as follows: $C(t) = t$, $C(id_{(u,x)}) = id_u \otimes id_x$, $C(d_{z,z}) = c_{z,z}$, $C(\beta' \otimes \beta'') = C(\beta') \otimes C(\beta'')$ and $C(\beta' * \beta'') = (C(\beta') \otimes u'); (v \otimes C(\beta''))$ if $\beta' : (u, x) \longrightarrow (v, y)$ and $\beta'' : (u', x') \longrightarrow (v', y')$. It can be verified that any different expression denoting β yields the same result. Moreover, if $\alpha : (u, 0) \longrightarrow (v, 0) \in \mathcal{CG}[B]/\Psi$ then the process obtained from $C(\alpha)$ by forgetting the label-indexed ordering functions of origins and destinations denotes a connected step of B.

Conversely, let $\xi = [\![\omega]\!]_\approx$ (for some causal firing sequence ω) be a connected step. The concatenable process $pr(\omega)$ of N_B can be denoted algebraically as the sequential and parallel composition of transitions, identities and symmetries. Moreover, we can take an equivalent process C without stable symmetries, i.e., C can be expressed as $\alpha_1; \ldots; \alpha_n$ where $\alpha_i = \beta_i \otimes u_i \otimes x_i$ with $u_i \in L_B^\oplus$, $x_i \in Z_B^\oplus$ and $\beta_i \in T_B \cup \{c_{kz,z}\}_{z \in Z_B, k \in \mathbb{N}}$. Then take $\alpha' = \alpha'_1 * \ldots * \alpha'_n$ where $\alpha'_i = \beta'_i \otimes x_i$ and $\beta'_i = \beta_i$ if $\beta_i \in T_B$ and $\beta'_i = d_{kz,z}$ if $\beta_i = c_{kz,z}$ for some zero place z, and integer k. Eventually, $\alpha = \alpha' \otimes u'$ where u' is the multiset of idle tokens.

This result states the correspondence among algebraic and operational semantics.

4.3 Abstract Semantics as Coreflection

Finally, we present the universal construction of the abstract semantics of ZS nets. We make use of a category **ZSC** whose objects are ZS nets and whose morphisms allow for the refinement of a transition into a connected transaction. This construction is somehow reminiscent of the construction of **ImplPetri** in [14].

Definition 17. Given ZS net B, a *causal abstract transition* of a $\mathcal{CG}[B]/\Psi$ is either a prime arrow of $\mathcal{CG}[B]/\Psi$ or a transition of B. Given two ZS net B and B', a *causal refinement morphism* $h : B \longrightarrow B'$ is a disjoint ZS net morphism $h = (f, g_L, g_Z)$ from B to (the image through the forgetful functor of) $\mathcal{CG}[B']/\Psi$ such that function f maps transitions into causal abstract transitions.

Since morphism h is disjoint, a transition can be refined into a transaction iff both its preset and its postset are stable. Transition involving zero places can only be mapped to transitions.

Lemma 18. *Given a causal refinement morphism $h : B \longrightarrow B'$, it uniquely extends to a morphisms $\hat{h} : \mathcal{CG}[B]/\Psi \longrightarrow \mathcal{CG}[B']/\Psi$ in* **ZSCGraph**, *which preserves prime arrows.*

Definition 19 (Category ZSC). The category **ZSC** has ZS nets as objects and causal refinement morphisms as arrows, their composition being defined through the extension in **ZSCGraph** given by Lemma 18.

Theorem 20. *Category* **Petri** *is embedded in* **ZSC** *fully and faithfully as a coreflective subcategory. Furthermore, the right adjoint of the coreflection $\mathcal{I}[\_]$ maps every ZS net B into its causal abstract net I_B (see Def. 5).*

Proof. (Sketch). The connected transactions (i.e. prime arrows, by Theorem 16) of a P/T net are all and only its transitions. Thus a causal refinement morphism $h : N \longrightarrow N'$ maps transitions into transitions. Next we want to prove that the obvious inclusion functor from **Petri** to **ZSC** has a right adjoint $\mathcal{I}[\_]$: **ZSC** $\longrightarrow$ **Petri** such that $\mathcal{I}[\_]$ maps each ZS net B into its causal abstract net I_B. We verify that $\mathcal{I}[\_]$ extends to a functor. Consider a causal refinement morphism $h = (f, g_L, g_Z) : B \longrightarrow B'$. Let $\hat{h} : \mathcal{CG}[B]/\Psi \longrightarrow \mathcal{CG}[B']/\Psi$ be the unique extension of h in **ZSCGraph**. Morphism $\hat{h}$ preserves prime arrows (by Lemma 18). Then we define $\mathcal{I}[h] = (f', g)$ with $f'(\xi) = \hat{h}(\xi)$ for any $\xi \in \Xi_B$ and $g(a) = g_L(a)$ for any $a \in L_B$. It follows that the unit component η_N of the adjunction is the identity and the counit component ϵ_B maps each transition of the abstract net into the appropriate connected transaction.

Category **ZSC** can be thought to represent the operational models, while **Petri** defines 'abstract' models. The functor $\mathcal{I}$: **ZSC** $\longrightarrow$ **Petri** that maps each ZS net B onto its abstract P/T net I_B, is the right adjoint to the inclusion functor. For every ZS net B there is a unique arrow $\epsilon_B : \mathcal{I}[B] \longrightarrow B$ with the universal property that, given any abstract model N in **Petri**, for every arrow $h : N \longrightarrow B$ there is a unique arrow $h' : N \longrightarrow \mathcal{I}[B]$ with $h = h'; \epsilon_B$. This situation is ideal from a semantic point of view. In fact $\mathcal{I}[B]$ can be understood as an abstraction of model B (e.g. its behaviour), with the additional advantage of being at the same time a model itself. The universal property above means that if we observe models from an abstract point of view (i.e. via morphisms originating from objects in **Petri**), then there is an isomorphism (via left composition with ϵ_B) between observations of B and observations of its abstract counterpart $\mathcal{I}[B]$. Thus in a sense, seen from **Petri**, B is the same as I_B.

5 Conclusion

We have proposed ZS nets as a model which offers the basis for a uniform approach to concurrent language translations. E.g., CCS-style languages may be easily modelled by representing the channels as zero places, in the style of our multicasting example. In this paper we have based our constructions on the so-called individual token philosophy [9]. Correspondingly, our categorical models rely on monoidal graphs equipped with an operation of horizontal composition together with a collection of special transitions called *swappings* to represent the permutations of tokens, along the style of [5]. An alternative and simpler approach corresponds to the so-called *collective token philosophy* as illustrated in [4]. We noticed that, whatever the adopted philosophy is, an identical restriction, called 'disjoint image property', must be required for the arrows of categories **ZSC** and **ZSN** (of [4]), which are used to define the abstract semantics. However, depending on the chosen approach – ITph vs CTph – two notion of transactions may be defined, each leading to different operational and abstract models. This should help to clarify the distinction between the two philosophies also from a pragmatic perspective rather than just from an academic viewpoint.

As a final remark, *symmetric, strict monoidal double categories* [3] seem to offer an alternative categorical characterization for the semantics of ZS nets. In this sense a ZS net could be viewed as a simple instance of a *tile rewrite system* [3, 7] where basic tiles are net transitions, and the horizontal composition of tiles corresponds to composition $\_ * \_$. The vertical composition of tiles would essentially build causal step sequences.

References

1. E. Best, R. Devillers and J. Hall. The Box Calculus: A New Causal Algebra with Multi-label Communication. In *Advances in Petri Nets '92, LNCS, n. 609*, 21–69. Springer-Verlag, 1992.
2. C. Brown and D. Gurr. A Categorical Linear Framework for Petri Nets. In *Proceedings of the 5th LICS Symposium*, 208–218, 1990.
3. R. Bruni, J. Meseguer, and U. Montanari. *Process and Term Tile Logic.* Technical Report, SRI International, to appear.
4. R. Bruni and U. Montanari. Zero-Safe Nets, or Transition Synchronization Made Simple. In *Proceedings of EXPRESS'97, ENTCS, Vol.7, 1997.*
5. P. Degano, J. Meseguer, and U. Montanari. Axiomatizing the Algebra of Net Computations and Processes. *Acta Informatica*, 33(7):641–667, October 1996.
6. P. Degano, R. De Nicola, and U. Montanari. A Distributed Operational Semantics for CCS based on Condition/Event Systems. *Acta Informatica*, 26:59–91, 1988.
7. F. Gadducci and U. Montanari. The Tile Model In: Gordon Plotkin, Colin Stirling, and Mads Tofte, Eds., *Proof, Language and Interaction: Essays in Honour of Robin Milner* MIT Press, to appear.
8. R. Gorrieri and U. Montanari. On the Implementation of Concurrent Calculi into Net Calculi: Two Case Studies *TCS 141, 1-2*, 1995, 195–252.
9. R.J. Van Glabbeek and G.D. Plotkin. Configuration Structures. In D. Kozen, editor, *Proceedings of the 10th LICS Symposium, IEEE*, pages 199–209, 1995.
10. R. Van Glabbeek and F. Vaandrager. Petri Net Models for Algebraic Theories of Concurrency. In *Proc. of PARLE, LNCS, n. 259*, 224–242. Springer-Verlag, 1987.
11. U. Goltz and W. Reisig. The Non-Sequential Behaviour of Petri Nets. *Information and Computation*, 57:125–147, 1983.

12. S. MacLane. *Categories for the Working Mathematician*. Springer-Verlag, 1971.
13. J. Meseguer and U. Montanari Petri Nets are Monoids: A New Algebraic Foundations for Net Theory. *Proc. 3rd LICS Symposium*, IEEE 1988:155–164.
14. J. Meseguer and U. Montanari. Petri Nets are Monoids. *Information and Computation*, 88(2):105–155, October 1990.
15. J. Meseguer, U. Montanari, and V. Sassone. Process versus Unfolding Semantics for Place/Transition Petri Nets. *TCS, Volume 153*, issue 1-2, (1996) pages 171-210.
16. J. Meseguer, U. Montanari, and V. Sassone. Representation Theorems for Petri Nets. Festschrift in honor of Prof. Wilfried Brauer to appear.
17. E.R. Olderog. Operational Petri Net Semantics for CCSP. In G. Rozenberg, editor, *Advances in Petri Nets '87, LNCS, n. 266*, 196–223. Springer-Verlag, 1987.
18. W. Reisig. *Petri Nets*. Springer-Verlag, 1985.
19. G. Ristori. *Modelling Systems with Shared Resources via Petri Nets* . PhD thesis TD 05/94, Department of Computer Science, University of Pisa, 1994.
20. V. Sassone. An Axiomatization of the Algebra of Petri Net Concatenable Processes. *Theoretical Computer Science*, vol. 170, n.1–2, pp 277–296, 1996.
21. G. Winskel. Event Structure Semantics of CCS and Related Languages. In *Proceedings of ICALP '82, LNCS, n. 140*, pages 561–567. Springer-Verlag, 1982.
22. G. Winskel. Petri Nets, Algebras, Morphisms and Compositionality. *Information and Computation*, 72:197–238, 1987.

A Symmetric, Strict Monoidal Categories

A *symmetric, strict monoidal category* [12], *ssmc* for short, is a quadruple $(\mathcal{C}, \otimes, e, \gamma)$ where $\mathcal{C}$ is the underlying category (with composition $\_;\_$ and identity id_x for each object x), functor $\otimes : \mathcal{C} \times \mathcal{C} \longrightarrow \mathcal{C}$ is the *tensor product*, object e of $\mathcal{C}$ is called the *unit object*, the diagrams

$$
\begin{tikzcd}
\mathcal{C} \times \mathcal{C} \times \mathcal{C} \arrow[r, "\otimes \times 1"] \arrow[d, "1 \times \otimes"'] & \mathcal{C} \times \mathcal{C} \arrow[d, "\otimes"] \\
\mathcal{C} \times \mathcal{C} \arrow[r, "\otimes"'] & \mathcal{C}
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathcal{C} \arrow[r, "\langle 1, e \rangle"] \arrow[dr, "1"'] & \mathcal{C} \times \mathcal{C} \arrow[d, "\otimes"] & \mathcal{C} \arrow[l, "\langle e, 1 \rangle"'] \arrow[dl, "1"] \\
 & \mathcal{C} &
\end{tikzcd}
$$

commute (where $\langle \_, \_ \rangle$ denotes the pairing of functors induced by the cartesian product of categories), and natural transformation $\gamma : \_{}_1 \otimes \_{}_2 \Rightarrow \_{}_2 \otimes \_{}_1$ is an isomorphism called *symmetry* satisfying the Kelly-MacLane *coherence axioms* $\gamma_{x \otimes y, z} = (id_x \otimes \gamma_{y,z}); (\gamma_{x,z} \otimes id_y)$, and $\gamma_{x,y}; \gamma_{y,x} = id_{x \otimes y}$ (for any objects x, y and z). A *symmetric strict monoidal functor* is a functor $F_\otimes : \mathcal{C} \longrightarrow \mathcal{C}'$ which preserves the monoidal structure and the symmetries.

Let **SSMC** be the category of symmetric strict monoidal categories and symmetric strict monoidal functors. We denote with $\mathbf{SSMC}^{\oplus}$ the full subcategory of **SSMC** consisting of the monoidal categories whose objects form *free* commutative monoids.