Electronic Notes in Theoretical Computer Science 7 (1997)
URL: http://www.elsevier.nl/locate/entcs/volume7.html 20 pages

Zero-Safe Nets, or Transition Synchronization Made Simple [1]

Roberto Bruni [2]

Computer Science Department
University of Pisa
Corso Italia 40, 56125 Pisa, Italy.

Ugo Montanari [3]

Computer Science Laboratory
SRI International
333 Ravenswood Ave., Menlo Park, CA 94025 USA

Abstract

In addition to ordinary places, called *stable*, zero-safe nets are equipped with *zero* places, which in a stable marking cannot contain any token. An evolution between two stable markings, instead, can be a complex computation called *stable transaction*, which may use zero places, but which is atomic when seen from stable places: no stable token generated in a transaction can be reused in the same transaction. Every zero-safe net has an ordinary Place-Transition net as its abstract counterpart, where only stable places are maintained, and where every transaction becomes a transition. The two nets allow us to look at the same system from both an abstract and a refined viewpoint. To achieve this result no new interaction mechanism is used, besides the ordinary token-pushing rules of nets. The refined zero-safe nets can be much smaller than their corresponding abstract P/T nets, since they take advantage of a transition synchronization mechanism. For instance, when transactions of unlimited length are possible in a zero safe net, the abstract net becomes infinite, even if the refined net is finite. In the second part of the paper two universal constructions - both following the *Petri nets are monoids* approach and the *collective token* philosophy - are used to give evidence of the naturality of our definitions. 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.

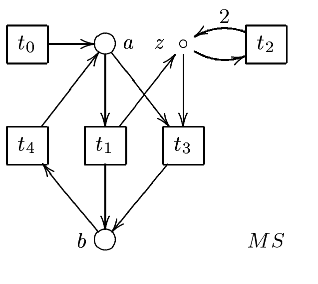
Key words: Petri nets are monoids. Abstraction. Transition synchronization. Transaction. Collective token philosophy.

1 Introduction

Petri nets [18,17], are unanimously considered one of the most attractive models of concurrency. As a matter of fact, this model offers a basic concurrent framework that has often been used as a semantic foundation on which to interpret concurrent languages (see for instance [19,10,16,5,8,2]). However the basic net model does not offer any synchronization mechanism among transitions, while this feature is essential to write modular, expressive programs. Thus all the above translations involve complex constructions for the net defining the synchronized composition of two programs.

In this paper a new kind of net is presented which offers a very general notion of transition synchronization as a built-in feature. More precisely, an abstract P/T net and a refined *zero-safe* net are supposed to model the same given system. The former offers the synchronized view and the latter specifies how every transition of the former is actually achieved as a coordinated collection of its transitions.

Zero-safe nets are based on the notion of *zero* places. Not all the places are zero places, however: the non-zero places are called *stable*. Stable markings (which consist only of stable tokens) describe the abstract-level markings, whilst non-stable markings define non-observable global states of the refined model. Thus a synchronized evolution of the zero-safe net (which we call *transaction*) starts at some observable marking, evolves through non-observable states and finally leads to a new observable state. No new interaction mechanism is used for building transactions, besides the ordinary token-pushing rules of nets. However, we do not associate an abstract transition to every transaction, but rather we take a concurrent view by identifying the transactions which are equivalent with respect to the usual diamond transformation. Thus the actual order of execution of concurrent transitions in the refined net is invisible in the abstract net.



$$t_0 = \mathit{new}$$
$$t_1 = \mathit{send}$$
$$t_2 = \mathit{copy}$$
$$t_3 = \mathit{receive}$$
$$t_4 = \mathit{reset}$$

Fig. 1. A zero-safe net representing a multicasting system.

[1] Research supported in part by Esprit Working Group CONFER2 and by Progetto Coordinato CNR '*Strumenti per la Verifica di Proprietà Critiche di Sistemi Concorrenti e Distribuiti*'.
[2] Corr. author: Phone: +39 50 887268, Fax: +39 50 887226, Email: `bruni@di.unipi.it`.
[3] On leave from University of Pisa, Computer Science Department.

To draw zero-safe nets, we extend the standard graphical representation for nets[4] by picturing zero places with smaller circles as in Fig. 1 (place z).

In what follows, we will use the zero-safe net MS of Fig. 1 as our running example. Places a and b are stable while place z is a zero place. Net MS is intended to represent a *multicasting system*. As in a broadcasting system, a process can simultaneously send the same message to an unlimited number of receivers, but here the receivers are not necessarily all the remaining processes, and thus several one-to-many communications can take place concurrently. We can interpret each token in place a as a different active process. To allow for an unlimited number of processes, the initial marking is empty, but tokens in place a can be created by the *new* transition t_0. A firing of transition t_1 (*send*) opens a one-to-many communication: the message is put in the buffer z and the process which started the communication is suspended until the end of the transaction. Each time the *copy* transition t_2 fires, a new copy of the same message is created. To complete a transaction, as many simultaneous occurrences of *receive* transition t_3 are needed, as the number of copies of the message created by *copy*, plus one. Each occurrence of *receive* synchronizes an active process (i.e. a token of a) with a copy of the message (i.e. a token of z). Transition *reset* (t_4) makes processes active again.

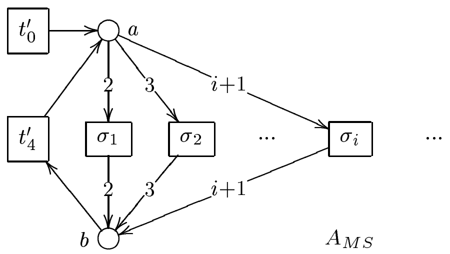


Fig. 2. The abstract net for the multicasting system in Fig. 1

In Fig. 2 we see an infinite P/T net representing the abstract net corresponding to the zero-safe net MS of Fig. 1. The abstract net A_{MS} comes equipped with a *refinement morphism* ϵ_{MS} to the refined net MS. In this case the morphism maps places a and b of A_{MS} into the homonymous stable places of MS. Furthermore, ϵ_{MS} maps each transition of A_{MS} into a transaction of MS. For instance, the transition σ_3 corresponds to a one-to-three transmission and is mapped into a transaction consisting of one instance of *send*, two instances of *copy* and three instances of *receive*. Actually there are two transactions of MS made that way:

- *send-copy-copy-receive-receive-receive*, and
- *send-copy-receive-copy-receive-receive*.

[4] Places and transitions are represented by circles and boxes respectively, each dot inside a place represents a token and directed weighted arcs describe the flow relation, omitting unitary weights by convention.

They differ for the order in which, after *send-copy*, the transitions *copy* and *receive* are executed. Notice that these transitions are concurrently enabled. Thus the two transactions are equal up to a diamond transformation, and transition σ_3 is more precisely mapped by ϵ_{MS} on their equivalence class.

The paper is organized as follows: after recalling the basic definitions of P/T nets, in Section 3 we introduce zero-safe nets and the corresponding abstract P/T nets. Section 4 has a more mathematical flavour. Its aim is to give evidence that the definitions and the constructions presented in the paper are natural, employing some elementary category theory. After some informal introduction of the categorical concepts involved, two universal constructions - both following the *Petri nets are monoids* approach and the *collective token* philosophy - are presented: the operational semantics of zero-safe nets is characterized as an adjunction, and the derivation of abstract P/T nets as a coreflection.

2 Place-Transition Nets

We introduce some basic definitions on nets.

Definition 2.1 *[Net] A* net *N is a triple $N = (S_N, T_N; F_N)$ where S_N is the (nonempty) set of* places $a, a', \ldots$, *T_N is the set of* transitions $t, t', \ldots$ *(with*[5] *$S_N \cap T_N = \emptyset$), and $F_N \subseteq (S_N \times T_N) \cup (T_N \times S_N)$ is called the* flow relation. $\square$

For $x \in N$, the set ${}^\bullet x = \{y \in N \mid yFx\}$ is called the *pre-set* of x, and the set $x^\bullet = \{y \in N \mid xFy\}$ is called the *post-set* of x. Analogously, for $X \subseteq N$ we define ${}^\bullet X = \bigcup_{x \in X} {}^\bullet x$ and $X^\bullet = \bigcup_{x \in X} x^\bullet$.

Place/transition nets are the most widespread model of nets. The places of a P/T net can hold one or more *tokens* and the arcs are weighted.

Definition 2.2 *[P/T Net] A* marked place/transition net *is a 5-tuple $N = (S, T; F, W, u_{\mathrm{in}})$ such that $(S, T; F)$ is a net, function $W : F \longrightarrow \mathbb{N}$ assigns a positive* weight *to each arc, and the multiset $u_{\mathrm{in}} : S \longrightarrow \mathbb{N}$ is the* initial marking *of N.* $\square$

In what follows we will sometimes refer to P/T nets simply as nets. The domain of the weight function can be extended to the whole $(S \times T) \cup (T \times S)$ by assuming $W(x, y) = 0$ when $(x, y) \notin F$. We find it convenient to interpret relation F as a function $F : ((S \times T) \cup (T \times S)) \longrightarrow \{0, 1\}$ with the convention that $xFy \iff F(x, y) \neq 0$. This allows to extend the theory to nets with weighted arrows simply replacing $\{0, 1\}$ by $\mathbb{N}$, throwing away W. Thus, relation $F : (S \times T) \cup (T \times S) \longrightarrow \mathbb{N}$ becomes a *multiset*[6] relation over

[5] In what follows, we will denote $S_N \cup T_N$ by N whenever no confusion is possible. Moreover, the index N is omitted from the terms S_N, T_N and F_N if it is obvious from the context.
[6] Given a multiset μ, we will use the notation $|\mu|$ to denote the set $\{a \mid \mu(a) > 0\}$ of elements included at least once in μ.

$(S \times T) \cup (T \times S)$ and moreover we can refer to the net N as the quadruple $(S, T; F, u_{\text{in}})$.

A *marking* $u : S \longrightarrow \mathbb{N}$ is a finite multiset of places. It can be written as $u = \{n_1 a_1, ..., n_k a_k\}$ where the natural number $n_i > 0$ indicates the number of occurrences (*tokens*) of the place a_i in u, i.e. $n_i = u(a_i)$. 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 $post(t)(a) = F(t,a)$ $\forall a \in S$. It follows from the definition that ${}^\bullet t = |pre(t)|$ and $t^\bullet = |post(t)|$.

The evolution of a net (i.e. its interleaving behaviour) is usually described in terms of firing sequences.

Definition 2.3 *[Enabling] Let N be a net and u a marking of N; then a transition $t \in T_N$ is* enabled at u *iff $pre(t)(a) \leq u(a)$, $\forall a \in S_N$.* □

Definition 2.4 *[Firing] Let N be a net, let u and u' be markings of N, and let t be a transition of N. 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 $u_0[t_1\rangle u_1 ... u_{n-1}[t_n\rangle u_n$. 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 also usually introduced. A step allows for the simultaneous execution of several independent transitions. Another important notion is *safety*. 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. $\forall u \in [u_{\text{in}}\rangle, \forall a \; u(a) \leq n$.

3 Zero-Safe Nets

We augment P/T nets with special places called zero places. Their role is to coordinate the atomic execution of complex collections of transitions, which can be considered as synchronized. However no new interaction mechanism is needed, and the coordination of the transitions participating in a step is handled by the ordinary token-pushing rules of nets.

Definition 3.1 *[ZS net] A* zero-safe[7] *net is a 6-tuple $B = (S_B, T_B; F_B, W_B, u_B; Z_B)$ where $N_B = (S_B, T_B; F_B, W_B, u_B)$ is the* underlying *P/T net and the set $Z_B \subseteq S_B$ is the set of* zero places *(also* 0-places *or* synchronization places*). The places in $S_B \setminus Z_B$ are called* stable places. *A* stable marking *is a multiset of stable places.* □

[7] In the standard terminology, a n-safe net is a net whose places are all n safe. Instead, in zero-safe nets only a subset of places (the zero places) are required to satisfy a 0-safe condition.

Stable markings describe *observable* states of the system, while the presence of one or more zero places in a given marking makes it unobservable.

A *stable step* of a zero-safe net B may involve the execution of several transitions of the underlying P/T net N_B (it is actually a firing sequence of N_B). There must be enough tokens on the stable (nonzero) places to enable all these transitions independently, while the tokens on zero places can be reused. However no token must be left on zero places at the end of the step (or can be found on them at the beginning of the step). Stable *transactions* are stable steps where no intermediate marking is stable and which consume all the available stable tokens. In a certain sense, each step can be thought of as a collection of transactions plus a collection of idle resources; this means that once you know what the possible transactions are, then you are able to construct all the *correct* behaviours of the system. We ask the reader to keep in mind this observation because it constitutes the basis for our approach. Stable *step sequences* are sequences of stable steps.

Definition 3.2 *[Stable step, transaction and step sequence] Let B be a zero-safe net and let $s = u_0[t_1\rangle u_1 \dots u_{n-1}[t_n\rangle u_n$ be a firing sequence of the underlying net N_B of B.*

Sequence s is a stable step *of B if:*

- $\forall a \in S_B \setminus Z_B,\ \sum_{i=1}^{n} pre(t_i)(a) \leq u_0(a)$ *(enabling);*
- *u_0 and u_n are stable markings of B (stable fairness).*

We write $u_0\{\!\![s\rangle u_n$ and $O(s) = u_0$, $D(s) = u_n$.

Stable step s is a stable transaction *of B if in addition:*

- *markings $u_1, \dots, u_{n-1}$ are not stable (atomicity);*
- $\forall a \in S_B \setminus Z_B,\ \sum_{i=1}^{n} pre(t_i)(a) = u_0(a)$ *(perfect enabling).*

A stable step sequence *is a sequence $u_0\{\!\![s_1\rangle u_1 \dots u_{n-1}\{\!\![s_n\rangle u_n$. We also say that u_n is* reachable from *u_0 and we write $u_n \in \{\!\![u_0\rangle$. Sometimes we will refer to the set $\{\!\![u_B\rangle$ of* reachable markings *of B with $\{\!\![B\rangle$.* □

In a stable transaction, each transition represents a micro-step carrying on the atomic evolution through invisible states. Stable tokens produced during the transaction become operative in the system only at the end of the transaction (i.e. after the firing of the *commit* transition t_n).

Example 3.3 *Consider the ZS net MS of Fig. 1.*

The firing sequence $\{a\}[t_1\rangle\{b,z\}[t_4\rangle\{a,z\}[t_3\rangle\{b\}$ is not a stable step since the enabling condition is not satisfied.

The firing sequence $\{4a\}[t_1\rangle\{3a,b,z\}[t_2\rangle\{3a,b,2z\}[t_3\rangle\{2a,2b,z\}[t_3\rangle\{a,3b\}$ is a stable step but not a stable transaction since the perfect enabling condition is not satisfied.

The firing sequence $s' = \{2a,b\}[t_1\rangle\{a,2b,z\}[t_3\rangle\{3b\}[t_4\rangle\{a,2b\}$ is a stable step but not a stable transaction since the atomicity constraint is not satisfied.

The firing sequence $s'' = \{2a, b\}[t_1\rangle\{a, 2b, z\}[t_4\rangle\{2a, b, z\}[t_3\rangle\{a, 2b\}$ *is a stable transaction.* □

Example 3.4 *Now consider the net* MS' *which is obtained from* MS *by deleting the transition* t_0 *and suppose that the actual marking consists of a token into place* a*. Then net* MS' *has no other reachable marking, because the only enabled transition is* t_1 *and we prevent from reactivating the process via* t_4 *until the transaction actually leaving a token in* z *is closed. If we relaxed this assumption then the system would allow communications between a process and itself.* □

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. In the case of P/T nets, the simplest approach is the *collective token* philosophy (see for instance [9]) which identifies all firing sequences obtained by repeatedly permuting pairs of firings which are concurrently (i.e. independently) enabled. An alternative approach, the *individual token* philosophy, will be discussed in the concluding remarks.

Definition 3.5 *[Diamond transformation, Abstract sequence] Given a P/T net* N*, let*

$$s = u_0[t_1\rangle u_1 \cdots u_{i-1}[t_i\rangle u_i[t_{i+1}\rangle u_{i+1} \cdots u_{n-1}[t_n\rangle u_n$$

be a firing sequence of N*. Now suppose that* t_i *and* t_{i+1} *are* concurrently enabled *by* u_{i-1}*, i.e.* $pre(t_i)(a) + pre(t_{i+1})(a) \leq u_{i-1}(a)$ *for any place* a*. Let* s' *be the firing sequence obtained by permuting the firing ordering of* t_i *and* t_{i+1}*, i.e.:*

$$s' = u_0[t_1\rangle u_1 \cdots u_{i-1}[t_{i+1}\rangle u'_i[t_i\rangle u_{i+1} \cdots u_{n-1}[t_n\rangle u_n.$$

The sequence s' *is a* diamond transformed *of* s*. The reflexive and transitive closure of the relation induced by diamond transformations gives the natural equivalence in the collective token interpretation. Notice that all the equivalent sequences have the same first and last markings* u_0 *and* u_n*. Equivalence classes are called* abstract sequences *and are denoted by* σ*. The abstract sequence of* s *is written* $[\![s]\!]$*. We also write* $pre([\![s]\!]) = O(s)$ *and* $post([\![s]\!]) = D(s)$ *to denote the origins and the destinations of* $[\![s]\!]$*, respectively.* □

Example 3.6 *In our running example, suppose that the current marking is* $\{a, b\}$*. If* t_4 *fires then a new token is produced into place* a*. A firing of* t_1 *consumes a token from place* a*. In the individual token 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 collective token 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*. Thus* t_1 *and* t_4 *may fire in any order always originating equivalent computations.* □

We can now apply the last definition to obtain a more satisfactory notion of stable firing and transaction.

Definition 3.7 *[Abstract stable step and transaction] Given a ZS net B, an* abstract stable step *is an abstract sequence $[\![s]\!]$ of the underlying net N_B, where s is a stable step. An* abstract stable transaction *is an abstract sequence of N_B which contains only stable transactions of B. We denote by Σ_B the set of all abstract stable transactions of B.* □

It is easy to see that our equivalence preserves stable steps [8] but not stable transactions. Thus, it is not enough to require s to be a stable transaction to make sure that $[\![s]\!]$ is an abstract stable transaction.

Example 3.8 *As a counterexample showing that stable transactions are not preserved by our equivalence, consider the net MS defined in Fig. 1 and the stable steps s' and s'' of Ex. 3.3. It is easy to verify that $[\![s']\!] = [\![s'']\!]$, since s'' is obtained from s' by a diamond transformation. However s'' is a stable transaction whereas s' is not. Thus $[\![s'']\!]$ is not a stable transaction.*

Conversely the firing sequence[9]

$$s = \{4a\}[t_1\rangle\{3a,b,z\}[t_2\rangle\{\ldots\}[t_2\rangle\{\ldots\}[t_3\rangle\{\ldots\}[t_3\rangle\{a,3b,z\}[t_3\rangle\{4b\}$$

defines an abstract stable transaction $[\![s]\!]$. In fact the stable transaction

$$\overline{s} = \{4a\}[t_1\rangle\{3a,b,z\}[t_2\rangle\{\ldots\}[t_3\rangle\{\ldots\}[t_2\rangle\{\ldots\}[t_3\rangle\{a,3b,z\}[t_3\rangle\{4b\}$$

is the unique diamond transformed of s (and vice versa). □

Since the basic execution steps of a system modelled via ZS nets consist of abstract stable transactions, it is natural to define a high-level description of such a model as a net whose transitions are abstract stable transactions.

Definition 3.9 *[Abstract Net] Let $B = (S_B, T_B; F_B, W_B, u_B; Z_B)$ be a ZS net. The net $A_B = (S_B \setminus Z_B, \Sigma_B; F, u_B)$, with $F(a,\sigma) = pre(\sigma)(a)$ and $F(\sigma,a) = post(\sigma)(a)$, is the* abstract net *of B, where we recall that $pre(\sigma)$ and $post(\sigma)$ yield the first and last marking of any stable transaction in the equivalence class σ.* □

Example 3.10 *Let MS be the ZS net of our running example and let $\emptyset$ be its initial marking. Consider the following firing sequences of the underlying net N_{MS} of MS:*

$$s_{new} = \{\}[t_0\rangle\{a\},$$
$$s_{res} = \{b\}[t_4\rangle\{a\},$$
$$s_1 = \{2a\}[t_1\rangle\{a,b,z\}[t_3\rangle\{2b\},$$
$$\ldots$$

[8] This property follows immediately since the diamond transformation preserves the enabling and stable fairness properties required by Def. 3.2.

[9] We omit some inner marking for a matter of space; however they are univocally determined and can be easily recovered.

$$s_i = \{(i+1)a\}[t_1\rangle\{ia, b, z\}[t_2\rangle \cdots [t_2\rangle\{ia, b, iz\}[t_3\rangle \cdots [t_3\rangle\{(i+1)b\},$$
$$\cdots$$

where s_i has $i-1$ firings of t_2 and i firings of t_3.

Then $\Sigma_{MS} = \{t'_0, t'_4, \sigma_1, \ldots, \sigma_i, \ldots\}$ with $t'_0 = [\![s_{new}]\!]$, $t'_4 = [\![s_{res}]\!]$ and $\sigma_i = [\![s_i]\!]$, for $i \geq 1$. The (infinite) abstract net of MS is (partially!) pictured in Fig. 2. This abstract net consists of two places and infinitely many transitions: one for creating a new active process, one for reactivating a process after a synchronization, and one for each possible multicasting communication involving i receivers. □

Zero places can be used to coordinate and synchronize in a single transaction any number of transitions of the refined net. Thus it may well happen that the refined net is finite while the abstract net is infinite. This is the case, for instance, of our running example, which models a multicasting system where a message can be delivered to an unlimited number of addresses. Notice also that the abstract and the refined net both rely on the same basic token-pushing mechanism to express their behaviour. This similarity is the key of the constructions described in the next section.

4 Universal Constructions

This section has a more abstract and mathematical flavour. Its aim is to give evidence that the definitions and the constructions presented in the previous section are natural. The tool we use is some elementary category theory. In particular, three concepts are useful here. The first notion is the *category of models* itself, where the objects are models (in our case zero-safe nets) and arrows represent some notion of simulation. The choice of arrows is very informative, since they complement and in a sense redefine [10] the meaning of models.

The second notion is *adjunction*, which is useful to characterize "natural" constructions. The typical scenario includes two categories C_1 and C_2 - where C_2 has more structure than C_1 - and a (usually obvious) forgetful functor $\mathcal{U} : C_2 \longrightarrow C_1$ which deletes the extra structure. It might happen that $\mathcal{U}$ has a left adjoint $\mathcal{F} : C_1 \longrightarrow C_2$. If this is the case, $\mathcal{F}$ represents the 'best' construction for adding the extra structure. In fact the left adjoint is unique (up to isomorphism) and satisfies a key universal property.

The third notion is *coreflection*, which is a special kind of adjunction. Here the scenario includes a category C and a subcategory C' of it. Category C represents the operational models, while C' defines certain 'abstract' models. In addition there is a functor $\mathcal{G} : C \longrightarrow C'$ whose left adjoint is the inclusion functor from C' to C. For every object u of C there is a unique arrow $\epsilon_u : \mathcal{G}(u) \longrightarrow u$ with the universal property that, given any abstract object a of

[10] E.g. isomorphic objects are often identified.

C', for every arrow $f : a \longrightarrow u$ there is a unique arrow $f' : a \longrightarrow \mathcal{G}(u)$ with $f = f'; \epsilon_u$, as the following (commuting) diagram illustrates.

$$\begin{tikzcd} u & \mathcal{G}(u) \arrow[l, "\epsilon_u"'] \\ a \arrow[u, "\forall f"] \arrow[ur, "\exists! f'"'] & \end{tikzcd}$$

This situation is ideal from a semantic point of view. In fact $\mathcal{G}(u)$ can be understood as an abstraction of model u (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 C'), then there is an isomorphism (via left composition with ϵ_u) between observations of u and observations of its abstract counterpart $\mathcal{G}(u)$. Thus in a sense, model u seen from C' is the same as $\mathcal{G}(u)$. Again, if a coreflection exists between C and C' with the inclusion as left adjoint then it is unique up to isomorphism.

In this section we describe two constructions involving zero-safe nets. The first construction starts from a category $ZPetri$ (where zero-safe nets are considered as programs) and exhibits an adjuction from it to a category $HCatZPetri$ consisting of some kind of machines, equipped with operations and transitions between states. It is proved that this adjunction corresponds to the token-pushing semantics of zero-safe nets defined in the previous section, in the sense that the transitions of the machine $\mathcal{Z}[B]$ corresponding to a zero-safe net B are exactly the abstract stable steps of B.

The second construction starts from a different category ZSN of zero-safe nets (which however is strictly related to $HCatZPetri$), having the ordinary category $Petri$ of P/T nets as a subcategory, and yields a coreflection corresponding exactly to the construction of the abstract net in Def. 3.9.

4.1 Petri Nets are Monoids

Petri net theory can be profitably developed within category theory. Among the existing approaches we mention [20,12,3]. We follow the approach initiated in [12] (other references are [13,4,14,15]). This approach focuses on the monoidal structure of Petri nets, where the monoidal operation means parallel composition. The basic observation is that a Petri net is just a graph where the set of nodes is a commutative monoid freely generated by the set of places.

Definition 4.1 *[Graph] A* graph *is a quadruple $G = (V, T, \partial_0, \partial_1)$ where V is the set of* nodes*, T is the set of* arcs *and $\partial_0, \partial_1 : T \longrightarrow V$ are two functions called* source *and* target*, respectively. We write $t : u \longrightarrow v$ with the obvious meaning, to shorten the notation. A morphism h from G to G' is a pair (f, g) of functions $f : T \longrightarrow T'$ and $g : V \longrightarrow V'$ such that $g(\partial_i(u)) = \partial'_i(f(u))$ for $i = 0, 1$. This, together with the obvious componentwise composition of morphisms, defines the category **Graph**.* □

Definition 4.2 *[Petri net] A (place/transition)* Petri net *is a graph where the* *arcs are called* transitions *and where the set of nodes is the free commutative monoid*[11] $S^{\oplus}$ *over a set of* places S *(thus* $\partial_0, \partial_1 : T \longrightarrow S^{\oplus}$*). A* Petri net morphism *is a graph morphism* $h = (f, g)$ *where* g *is required to be a monoid homomorphism (i.e. leaving 0 fixed and respecting the monoid operation* $\oplus$*). This defines a category* ***Petri****.* □

In [13,4] it has been shown that it is possible to enrich the algebraic structure of transitions in order to capture some basic constructions on nets – case graphs, firing sequences, Goltz-Reisig and Best-Devillers processes ([11,1]), etc. In particular, in [13] a chain of adjunctions is defined, each adjunction showing a further enrichement of the algebraic structure on transitions. We are mainly interested in the definition of the category $CMonRPetri$.

Definition 4.3 *[Reflexive Petri Commutative Monoid] A* reflexive Petri commutative monoid M *is a Petri net together with a function* $id : S^{\oplus} \longrightarrow T$*, where:*

- *transitions form a commutative monoid* $(T, \otimes, 0)$*, and*
- *mappings* ∂_0*,* ∂_1 *and* id *are monoid homomorphisms.*

A reflexive Petri monoid morphism *is a Petri net morphism* $h = (f, g) : M \longrightarrow M'$ *preserving identities (i.e.,* $f(id_u) = id'_{g(u)}$*) and monoidal structures. This defines the category* $CMonRPetri$*.* □

The forgetful functor from $CMonRPetri$ to $Petri$ has a left adjoint which associates to each Petri net N its *marking graph* $\mathcal{C}[N]$. The following rules inductively define the arrows of $\mathcal{C}[N]$:

$$\frac{u \in S_N^{\oplus}}{id_u : u \longrightarrow u \in \mathcal{C}[N]} \qquad \frac{t : u \longrightarrow v \in T_N}{t : u \longrightarrow v \in \mathcal{C}[N]}$$

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

where the following equations, stating that $\mathcal{C}[N]$ is a reflexive commutative monoid, are to be satisfied (for all arrows α, β and γ and for all multisets u and v):

$$id_0 \otimes \alpha = \alpha,$$
$$(\alpha \otimes \beta) \otimes \delta = \alpha \otimes (\beta \otimes \delta),$$
$$\alpha \otimes \beta = \beta \otimes \alpha, \text{ and}$$
$$id_u \otimes id_v = id_{u \oplus v}.$$

[11] The elements of $S^{\oplus}$ will be presented as formal sums $n_1 a_1 \oplus \ldots \oplus n_k a_k$ with the order of summands being immaterial, $a_i \in S$ and $n_i \in I\!N$ for $i = 1, \ldots, n$. Moreover 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. Obviously, a marking $u = \{n_1 a_1, ..., n_k a_k\}$ just corresponds to the element $n_1 a_1 \oplus \ldots \oplus n_k a_k$ of $S^{\oplus}$.

Intuitively, the monoidal operator allows for the concurrent execution of transitions, and the identity function can be used to explicitely represent idle tokens. The marking graph $\mathcal{C}[N]$ corresponds to the ordinary operational semantics of N, i.e. its transitions are the step sequences of N.

4.2 Operational Semantics of Zero-Safe Nets

We now present the universal construction yielding the operational semantics of our nets. We first define the category of zero-safe nets.

Definition 4.4 *[Category* ***ZPetri****] 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* g *is a monoid homomorphism which preserves partitioning of places (i.e., if* $a \in Z$ *then* $g(a) \in Z'^{\oplus}$ *and if* $a \in S \setminus Z$ *then* $g(a) \in (S' \setminus Z')^{\oplus}$*). This defines a category* ***ZPetri****.* □

Remark 4.5 *Since* $S^{\oplus}$ *is a free commutative monoid we can equivalently represent the set of nodes of a ZS net as* $L^{\oplus} \times Z^{\oplus}$ *(i.e., sources and targets are pairs whose components are elements of the free commutative monoids over stable and zero places respectively). Thus ZS net morphisms become triples of the form* $h = (f, g_L, g_Z)$ *where both* g_L *and* g_Z *are monoid homomorphisms on the monoids of stable and zero places respectively.*

Example 4.6 *The graph corresponding to the ZS net* MS *defined in Fig. 1 has the following set of arcs:*

$$\begin{aligned} T_{MS} = \{ & t_0 : (0,0) \longrightarrow (a,0), t_1 : (a,0) \longrightarrow (b,z), t_2 : (0,z) \longrightarrow (0,2z), \\ & t_3 : (a,z) \longrightarrow (b,0), t_4 : (b,0) \longrightarrow (a,0)\}. \end{aligned}$$

□

We now introduce for zero-safe nets the category $HCatZPetri$ corresponding to $CMonRPetri$. However the models of $HCatZPetri$ (which we call *ZS graphs*) are more complex than those of $CMonRPetri$ since they must be equipped with an operation of composition of arrows to allow for the construction of transactions. Thus $HCatZPetri$ is in a sense intermediate between $CMonRPetri$ and the category $CatPetri$ introduced in [13].

Definition 4.7 *[Category* $HCatZPetri$*] A* ZS graph $H = ((L \cup Z)^{\oplus}, (T, \otimes, 0, id, ;), \partial_0, \partial_1)$ *is both a ZS net and a reflexive Petri commutative monoid. In addition, it is equipped with a partial function* $\_;\_$ *called* horizontal composition*:*

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

Horizontal composition is associative and has identities $id_{(0,x)} : (0,x) \longrightarrow (0,x)$ *for any* $x \in Z^{\oplus}$*. In addition, the commutative monoidal operator* $\_ \otimes \_$ *is functorial w.r.t. horizontal composition, i.e.*

$$(\alpha \otimes \beta);(\alpha' \otimes \beta') = (\alpha;\alpha') \otimes (\beta;\beta')$$

whenever the right member is defined. Given two ZS graphs H and H', a ZS graph morphism $h = (f, g_L, g_Z) : H \longrightarrow H'$ *is both a ZS net morphism and a reflexive Petri monoid morphism such that $f(\alpha;\beta) = f(\alpha); f(\beta)$. ZS graphs and horizontal morphisms (together with the obvious composition and identities) constitute the category HCatZPetri.* □

Horizontal composition is the key of our approach. It acts as a sequential composition on zero places and as a parallel composition on stable places. This is exactly what we need to model stable steps, because two successive firings in a stable step are allowed iff the stable tokens which are needed are already present in the initial marking.

Proposition 4.8 *If $\alpha : (u, 0) \longrightarrow (v, 0)$ and $\alpha' : (u', 0) \longrightarrow (v', 0)$ are two transitions of a ZS graph then $\alpha; \alpha' = \alpha \otimes \alpha'$.*

Proof. It can be easily verified that:

$$\alpha; \alpha' = (\alpha \otimes id_{(0,0)}); (id_{(0,0)} \otimes \alpha') = (\alpha; id_{(0,0)}) \otimes (id_{(0,0)}; \alpha') = \alpha \otimes \alpha'.$$

Moreover, since the monoidal operator $\_ \otimes \_$ is commutative, it follows that $\alpha; \alpha' = \alpha'; \alpha$. □

Next results show that $HCatZPetri$ has $CMonRPetri$ as a subcategory.

Proposition 4.9 *The full subcategory of HCatZPetri whose objects are all and only Petri nets (i.e., $Z = \emptyset$) is isomorphic to CMonRPetri.*

Proof. Let H be a ZS graph such that Z_H is empty. Then $\forall \alpha \in T_H$, there exist $u, v \in L_H^{\oplus}$ such that $\alpha : (u, 0) \longrightarrow (v, 0)$. Thus $\forall \alpha, \beta \in H$ it follows that $\alpha; \beta = \alpha \otimes \beta$, i.e. the horizontal composition adds no structure. □

The following theorem defines the algebraic semantics of zero-safe nets by means of a universal property.

Theorem 4.10 *Let $\mathcal{U} : HCatZPetri \longrightarrow ZPetri$ be the functor which forgets about the additional structure on transitions, i.e.*

$$\mathcal{U}[((L \cup Z)^{\oplus}, (T, \otimes, 0, id, ;), \partial_0, \partial_1)] = (L^{\oplus} \times Z^{\oplus}, T, \partial_0, \partial_1).$$

Functor $\mathcal{U}$ has a left adjoint $\mathcal{Z} : ZPetri \longrightarrow HCatZPetri$ which maps a ZS net B into the ZS graph defined by the following inference rules and axioms:

$$\frac{(u, x) \in L_B^{\oplus} \times Z_B^{\oplus}}{id_{(u,x)} : (u, x) \longrightarrow (u, x) \in \mathcal{Z}[B]} \qquad \frac{t : (u, x) \longrightarrow (v, y) \in T_B}{t : (u, x) \longrightarrow (v, y) \in \mathcal{Z}[B]}$$

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

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

Where transitions form a commutative monoid, i.e.

$$\alpha \otimes \beta = \beta \otimes \alpha,$$

$$(\alpha \otimes \beta) \otimes \delta = \alpha \otimes (\beta \otimes \delta), \ \textit{and}$$
$$id_{(0,0)} \otimes \alpha = \alpha,$$

for any $\alpha, \beta, \delta \in \mathcal{Z}[B]$; the horizontal composition operator $\_;\_$ is associative and has identities, i.e.

$$(\alpha; \beta); \delta = \alpha; (\beta; \delta), \ \textit{and}$$
$$\alpha; id_{(0,y)} = \alpha = id_{(0,x)}; \alpha$$

whenever such compositions are defined; finally the monoidal operator $\_ \otimes \_$ is functorial, this means that

$$id_{(u,x)} \otimes id_{(v,y)} = id_{(u \oplus v, x \oplus y)}, \ \textit{and}$$
$$(\alpha \otimes \alpha'); (\beta \otimes \beta') = (\alpha; \beta) \otimes (\alpha'; \beta')$$

(the latter holds whenever the rightmost member of the equation is defined).

Proof (Sketch) It is easy to verify that mapping $\mathcal{Z}$ extends to a functor which is a right adjoint to functor $\mathcal{U}$. □

The following theorem shows that the algebraic semantics of zero-safe nets is an extension of the ordinary semantics of P/T nets.

Theorem 4.11 *When restricted to P/T nets, functor $\mathcal{Z}$ coincides with $\mathcal{C}$.*

Proof. Immediately follows from Prop. 4.8 since if $Z_B = \emptyset$ then $\forall t : (u,x) \longrightarrow (v,y) \in T_B$ is $x = y = 0$. □

Example 4.12 *Let MS be the zero-safe net of our running example whose set of arcs is defined in Ex. 4.6. For instance the arrow $t_1; t_3 \in \mathcal{Z}[MS]$ 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 $\mathcal{Z}[MS]$, i.e., they all denote the same arrow:

$$t_1; t_2; (t_2 \otimes id_{(0,z)}); (t_3 \otimes id_{(0,2z)}); (t_3 \otimes id_{(0,z)}); t_3,$$
$$t_1; t_2; (t_2 \otimes id_{(0,z)}); (t_3 \otimes t_3 \otimes t_3),$$
$$t_1; t_2; (t_2 \otimes id_{(0,z)}); (id_{(0,2z)} \otimes t_3); (t_3 \otimes t_3), \ \textit{and}$$
$$t_1; t_2; (t_2 \otimes t_3); (t_3 \otimes t_3).$$

□

Next theorem shows that the operational and algebraic semantics of zero-safe nets coincide. We first need a definition.

Definition 4.13 *[Prime Transition] A transition $\alpha : (u,0) \longrightarrow (v,0)$ of a ZS graph H 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 $\alpha = \beta \otimes \gamma$).* □

Theorem 4.14 *Given a ZS net B, there is a one-to-one correspondence between arrows $\alpha : (u,0) \longrightarrow (v,0) \in \mathcal{Z}[B]$ and abstract stable steps of B.*

Moreover, if such an arrow is prime then the corresponding abstract stable step is an abstract stable transaction.

Proof (Sketch) Given (the equivalence class of) a generic stable step

$$s = u \oplus u_0[t_1\rangle u \oplus u_1 \oplus x_1[t_2\rangle \cdots u \oplus u_{n-1} \oplus x_{n-1}[t_n\rangle u \oplus u_n$$

where each multiset x_i contains (all) the zero tokens at the i-th stage of the step, u is the multiset of the stable tokens which are idle in s and $t_i : (w_i, y_i) \longrightarrow (v_i, z_i)$ for $i = 1, \ldots, n$. Then the corresponding arrow is

$$\alpha_s = (t_1 \otimes id_{(u,x'_1)}); (t_2 \otimes id_{(0,x'_2)}); \cdots; (t_n \otimes id_{(0,x'_n)})$$

with $x'_1 = 0$ and $x'_i \oplus y_i = x_{i-1}$, for $i = 2, \ldots, n$, where y_i is the multiset of zero places in the source of t_i (see above). The correctness of our definition follows immediately, simply noticing that each diamond transformed s' of s is mapped into an arrow $\alpha_{s'}$ which can be proved equal to α_s thanks to the functoriality axiom. In fact, since s' is a diamond transformed of s, then $\exists k$ such that t_k and t_{k+1} are concurrently enabled, i.e. $y_{k+1} \leq x'_k$. Thus (for generic stable markings v and v'):

$$(t_k \otimes id_{(v,x'_k)}); (t_{k+1} \otimes id_{(v',x'_{k+1})}) = t_k \otimes t_{k+1} \otimes id_{(v \oplus v',x')} =$$
$$= (t_{k+1} \otimes id_{(v',x' \oplus y_k)}); (t_k \otimes id_{(v,x' \oplus z_{k+1})})$$

where $x' \oplus y_{k+1} = x'_k$. Then, it can be easily checked that the axioms given in the proof of Theorem 4.10 identify equivalent steps only.

For the converse correspondence, let

$$(t_1 \otimes id_{(u_1,x_1)}); (t_2 \otimes id_{(u_2,x_2)}); \cdots; (t_n \otimes id_{(u_n,x_n)})$$

be any (arbitrarily chosen) *linearization* of a given α, with $t_j : (w_j, y_j) \longrightarrow (v_j, z_j)$. Then, the sequence

$$s = u'_0[t_1\rangle u'_1 \cdots u'_{k-1}[t_k\rangle u'_k[t_{k+1}\rangle u'_{k+1} \cdots u'_{n-1}[t_n\rangle u'_n$$

with $u'_0 = u$ is a stable step.

Now suppose that α is prime and that $[\![s]\!]$ is not a stable transaction. Then $\exists s' \in [\![s]\!]$ such that $s' = u[t_{i_1}\rangle p_1 \ldots p_{n-1}[t_{i_n}\rangle v$ with $p_k \in L_B^{\oplus}$ for a certain index k ($\sum_{j=1}^{n} pre(t_{i_j})(a) = u(a)$ for any stable place a, because α is prime). Then $\alpha = \beta; \gamma$ with $\beta : (q, 0) \longrightarrow (q', 0)$ and $\gamma : (r, 0) \longrightarrow (r', 0)$ for some (non-trivial) arrows β and γ with $u = q \oplus r$, $v = q' \oplus r'$ and $q' \oplus r = p_k$. This is contradictory, since $\alpha = \beta; \gamma = \beta \otimes \gamma$ while α is prime by hypothesis. □

Example 4.15 *In our running example the prime arrows of $\mathcal{Z}[MS]$ are*

$$\tau_0 = t_0,$$

$$\tau_4 = t_4,$$

$$\alpha_1 = t_1; t_3,$$

$$\alpha_2 = t_1; t_2; (t_3 \otimes id_{(0,z)}); t_3,$$

$$\ldots$$

$$\alpha_i = t_1; \beta_i; \delta_i, \quad with \begin{cases} \beta_i = t_2; (t_2 \otimes id_{(0,z)}); \cdots; (t_2 \otimes id_{(0,(i-2)z)}) \\ \delta_i = (t_3 \otimes id_{(0,(i-1)z)}); \cdots; (t_3 \otimes id_{(0,z)}); t_3 \end{cases}$$
$$\cdots$$

The correspondence with the abstract stable transactions of MS which are given in Ex. 3.10 is the intuitive one. As a further example, some more compact notations to define arrows α_i are either $\delta_i = t_3 \otimes \cdots \otimes t_3$ where t_3 is repeated exactly i times or $\alpha_i = t_1; t_2; (t_2 \otimes t_3); \cdots; (t_2 \otimes t_3); (t_3 \otimes t_3)$ where expression $(t_2 \otimes t_3)$ appears exactly $i - 2$ times. □

4.3 Abstraction of Zero-Safe Nets

We now present the universal construction yielding the abstract semantics of our nets. To this purpose we define a category ZSN of zero-safe nets where the morphisms may map a transition into a transaction. In essence, ZSN has the objects of $ZPetri$ and some of the arrows of $HCatZPetri$. This construction is reminiscent of the construction of $ImplPetri$ in [13].

Definition 4.16 *[Abstract Transition] An* abstract transition *of a given ZS net B is either a prime arrow of $\mathcal{Z}[B]$ or a transition of B.* □

Definition 4.17 *[Refinement Morphism] Given two ZS nets $B, B' \in ZPetri$, a* refinement morphism *$h : B \longrightarrow B'$ is a ZS net morphism $(f, g_L, g_Z) : B \longrightarrow \mathcal{Z}[B']$ such that function f maps transitions into abstract transitions and morphism $g_Z : Z_B^{\oplus} \longrightarrow Z_{B'}^{\oplus}$ maps zero places of B into pairwise disjoint (non-empty) elements[12] of $Z_{B'}^{\oplus}$.* □

Lemma 4.18 *Given a refinement morphism $h : B \longrightarrow B'$, let $\widetilde{h}$ be its unique extension in $HCatZPetri$. Then, morphism $\widetilde{h}$ preserves prime arrows.*

Proof (Sketch) We want to show that if α is prime in $\mathcal{Z}[B]$, then also $\widetilde{h}(\alpha)$ is prime in $\mathcal{Z}[B']$. Now let $u[t_1\rangle u_1 \cdots u_{n-1}[t_n\rangle u_n$ be a firing sequence corresponding to (a linearization of) α.

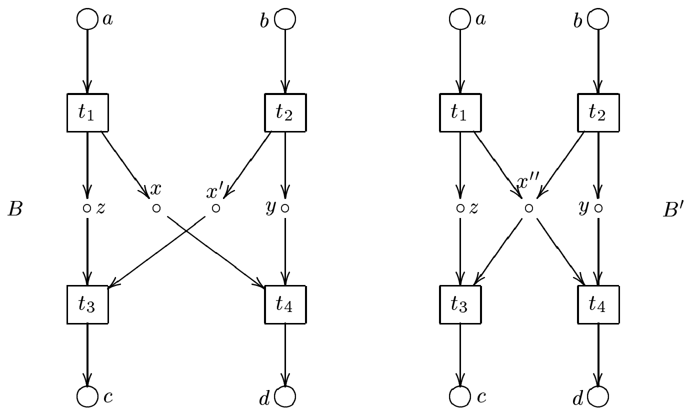
If $n = 1$ then $\widetilde{h}(\alpha) = \widetilde{h}(t_1) = h(t_1)$ which is prime.

If $n > 1$ then we proceed by contradiction. Suppose that $\widetilde{h}(\alpha)$ is not prime; this implies that $\exists \beta_1, \beta_2 \in \mathcal{Z}[B']$ with $\widetilde{h}(\alpha) = \beta_1 \otimes \beta_2$. Since α is prime, then each t_i involves at least a zero place. It follows, by Def. 4.16, that each $h(t_i)$ is a transition. This induces a corresponding linearization for $\widetilde{h}(\alpha)$, given by $h(u)[h(t_1)\rangle h(u_1) \cdots h(u_{n-1})[h(t_n)\rangle h(u_n)$. Moreover let $v[s_1\rangle v_1 \cdots v_{k-1}[s_k\rangle v_k$ and $w[s_{k+1}\rangle w_1 \cdots w_{l-1}[s_n\rangle w_l$ (with $k + l = n$) be some firing sequences corresponding to β_1 and β_2 respectively. Now suppose that a diamond transformation at position i can be applied to $\widetilde{h}(\alpha)$, i.e. $h(t_i)$ and $h(t_{i+1})$ are both enabled at $h(u_{i-1})$. The disjoint image property of h allows to infer that also t_i and t_{i+1} are both enabled at u_{i-1}, so a diamond transformation can also

[12] I.e. $\forall z, z' \in Z_B$ with $z \neq z'$, if $g_Z(z) = n_1 a_1 \oplus \cdots \oplus n_k a_k$ and $g_Z(z') = m_1 b_1 \oplus \cdots \oplus m_l b_l$ then we have that $a_i \neq b_j$, for $i = 1, \ldots, k$ and $j = 1, \ldots, l$.

be applied to α at position i. Iteratively applying to α the *specular* diamond transformations needed to reach the sequence $h(u)[s_1\rangle u''_1 \cdots u''_{n-1}[s_n\rangle u''_n$ starting from $\widetilde{h}(\alpha)$, we obtain the sequence $u[t_{i_1}\rangle u'_1 \cdots u'_{n-1}[t_{i_n}\rangle u'_n$ with $u'_n = u_n$ and where $h(u'_j) = u''_j$ and $h(t_{i_j}) = s_j$ for $j = 1, \ldots, n$. It is easy to show that the sequences $v'[t_{i_1}\rangle v'_1 \cdots v'_{k-1}[t_{i_k}\rangle v'_k$ and $w'[t_{i_{k+1}}\rangle w'_1 \cdots w'_{l-1}[t_{i_n}\rangle w'_l$ (with $v' \oplus w' = u$, $h(v') = v$, $h(v'_j) = v_j$ for $j = 1, \ldots, k$, $h(w') = w$ and $h(w'_j) = w_j$ for $j = 1, \ldots, l$) define two arrows α_1 and α_2 such that $\alpha = \alpha_1 \otimes \alpha_2$, thus contradicting the hypothesis that α is prime. $\square$

Remark 4.19 *The disjoint image property on zero places required for the morphisms in Def. 4.17 is necessary for Lemma 4.18 to hold. As an example, consider ZS nets B and B' pictured below:*



with

$$
\begin{aligned}
S_B &= \{a, b, c, d, z, x, x', y\}, \quad Z_B = \{z, x, x', y\},\\
T_B &= \{t_1 : (a, 0) \longrightarrow (0, z \oplus x), t_2 : (b, 0) \longrightarrow (0, x' \oplus y),\\
&\qquad t_3 : (0, z \oplus x') \longrightarrow (c, 0), t_4 : (0, x \oplus y) \longrightarrow (d, 0)\},\\
S_{B'} &= \{a, b, c, d, z, x'', y\}, \quad Z_{B'} = \{z, x'', y\},\\
T_{B'} &= \{t_1 : (a, 0) \longrightarrow (0, z \oplus x''), t_2 : (b, 0) \longrightarrow (0, x'' \oplus y),\\
&\qquad t_3 : (0, z \oplus x'') \longrightarrow (c, 0), t_4 : (0, x'' \oplus y) \longrightarrow (d, 0)\}.
\end{aligned}
$$

Then mapping $h : B \longrightarrow B'$, merging both zero places x and x' into x'' and leaving unchanged the rest, maps prime arrow $\alpha = (t_1 \otimes t_2); (t_3 \otimes t_4) \in \mathcal{Z}[B]$ into arrow $h(\alpha) = (t_1 \otimes t_2); (t_3 \otimes t_4) = (t_1; t_3) \otimes (t_2; t_4)$ which is not prime. In fact mapping h does not respect the disjoint image property.

Definition 4.20 *[Category ZSN] The category ZSN has ZS nets as objects and refinement morphisms as arrows. The composition between two refinement morphisms $h : B \longrightarrow B'$ and $h' : B' \longrightarrow B''$ is defined as the ZS net morphism[13] $\widetilde{h'} \circ h : B \longrightarrow \mathcal{Z}[B'']$, where $\widetilde{h'}$ is the unique extension of h' to a morphism in HCatZPetri.* $\square$

Theorem 4.21 *Category Petri is embedded into ZSN* fully *and* faithfully *as a coreflective subcategory. Furthermore the functor $\mathcal{A}[\_]$, which is the right*

[13] Lemma 4.18 guarantees that $\widetilde{h'} \circ h$ is a refinement morphism.

adjoint of the coreflection, maps every ZS net B into its abstract net A_B (see Def. 3.9), i.e. $\mathcal{A}[B] = A_B$.

Proof (Sketch) We start by defining the functor $\mathcal{D}[\_] : \mathit{Petri} \longrightarrow \mathit{ZSN}$. Let $\mathcal{D}[(S^\oplus, T, \partial_0, \partial_1)] = (S^\oplus \times \{0\}, T, (\partial_0, 0), (\partial_1, 0))$, i.e. $\mathcal{D}[N]$ is the ZS net generated by N whose nodes are renamed as pairs having the second component equal to 0. The abstract stable transactions (i.e. prime arrows, by Theorem 4.14) of $\mathcal{D}[N]$ are all and only its transitions. Thus a refinement morphism $h : \mathcal{D}[N] \longrightarrow \mathcal{D}[N']$ maps transitions into transitions. We extend $\mathcal{D}[\_]$ to a functor by defining $\mathcal{D}[(f, g)] = (f, g, 0)$. Next we want to prove that $\mathcal{D}[\_] \dashv \mathcal{A}[\_] : \mathit{ZSN} \longrightarrow \mathit{Petri}$ where $\mathcal{A}[\_]$ maps each ZS net B into its *abstract net* A_B. Consider a refinement morphism $h(f, g_L, g_Z) : B \longrightarrow B'$. Let $\widetilde{h}$ be the unique extension of h in $\mathit{HCatZPetri}$. Morphism $\widetilde{h}$ preserves prime arrows (by Lemma 4.18). Then mapping $\mathcal{A}[\_]$ extends to a functor by defining $\mathcal{A}[h] = (f', g_L)$ with $f'(\sigma) = \widetilde{h}(\sigma)\ \forall \sigma \in \Sigma_B$. It follows that the unit component η_N of the adjunction is the identity and the counit component ϵ_B maps transitions of the abstract net into appropriate abstract transactions. □

5 Conclusion and Future Work

In this paper we have based our constructions on the so-called collective token philosophy [9]. In fact we have defined abstract stable steps as the direct quotient of diamond-equivalent classes of stable steps. Correspondingly, we have based our categorical models on graphs equipped with a monoidal operation which is commutative on both nodes and arcs. As shown in [4], an alternative approach to the semantics of Petri nets introduces special transitions called *symmetries* to represent the permutations of tokens all present at the same place. Correspondingly, the categorical semantics is given in terms of (a suitable subclass of) *symmetric monoidal categories* (*ssmc*). In *ssmc*'s, the monoidal operation is not commutative and a natural transformation builds the symmetries. This alternative approach corresponds to the so-called *individual token philosophy* and offers a much more informative semantics. For instance, let us consider the net in Fig. 1 (the distinction between stable and zero places is immaterial here). If we execute several simultaneous one-to-many communications, the equivalence classes of firing sequences implied by the individual token approach fully distinguish which receiver has been synchronized with which sender [14]. This is not the case for the collective token approach, which just records the total number of senders and the total number of receivers.

In this paper we have followed the collective token philosophy for the sake of simplicity. In fact both the operational and the algebraic semantics turn out considerably simpler. However we believe that an individual token semantics

[14] Actually different copy policies for messages, e.g. sequential copying or balanced-tree copying, are also distinguished.

of zero-safe nets could be given without too much effort. We anticipate that certain restrictions we need here (for the arrows of category ZSN) could be possibly lifted in the individual token case.

Finally we want to mention a connection between zero-safe nets and the tile model [6,7]. Tiles are rewrite rules, similar to SOS inference rules, equipped with three operations of composition: horizontal, vertical and parallel. Horizontal composition builds tiles corresponding to synchronized steps, vertical composition to sequentialized steps and parallel composition to concurrent steps. Tiles can be exactly interpreted as double cells of a monoidal double category, and provide an expressive and clean metalanguage to define a variety of models of computation. Zero-safe nets represent the simple case where basic tiles are net transitions, and where the horizontal composition of tiles corresponds to the horizontal composition of arrows in the category $HCatZPetri$. The vertical composition of arrows would approximately correspond to building stable step sequences, as defined in Def. 3.2.

References

[1] E. Best and R. Devillers. Sequential and Concurrent Behaviour in Petri Net Theory. *Theoretical Computer Science*, 55:87–136, 1987.

[2] 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.

[3] C. Brown and D. Gurr. A Categorical Linear Framework for Petri Nets. In *Proceedings of the 5th LICS Symposium*, 208–218, 1990.

[4] P. Degano, J. Meseguer, and U. Montanari. Axiomatizing the Algebra of Net Computations and Processes. *Acta Informatica*, 33(7):641–667, October 1996.

[5] 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.

[6] F. Gadducci. *On the Algebraic Approach to Concurrent Term Rewriting.* PhD thesis TD 02/96, Department of Computer Science, University of Pisa, 1996.

[7] F. Gadducci and U. Montanari. The Tile Model. Technical Report TR-96-27, Department of Computer Science, University of Pisa, 1996.

[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 *Proceedings of PARLE, LNCS, n. 259*, pages 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] J. Meseguer and U. Montanari Petri Nets are Monoids: A New Algebraic Foundations for Net Theory. *Proc. 3rd Symp. on Logics in Computer Science*, IEEE 1988:155–164.

[13] J. Meseguer and U. Montanari. Petri Nets are Monoids. *Information and Computation*, 88(2):105–155, October 1990.

[14] 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.

[15] J. Meseguer, U. Montanari, and V. Sassone. Representation Theorems for Petri Nets. Festschrift in honor of Prof. Wilfried Brauer to appear.

[16] E.R. Olderog. Operational Petri Net Semantics for CCSP. In G. Rozenberg, editor, *Advances in Petri Nets '87, LNCS, n. 266*, pages 196–223. Springer-Verlag, 1987.

[17] C.A. Petri. *Kommunikation mit Automaten*. PhD thesis, Institut für Instrumentelle Mathematik, Bonn, Germany, 1962.

[18] W. Reisig. *Petri Nets*. Springer-Verlag, 1985.

[19] G. Winskel. Event Structure Semantics of CCS and Related Languages. In *Proceedings of ICALP '82, LNCS, n. 140*, pages 561–567, Springer-Verlag, 1982.

[20] G. Winskel. Petri Nets, Algebras, Morphisms and Compositionality. *Information and Computation*, 72:197–238, 1987.