

264956-3

INF

FL
2936

DEPARTAMENTO
DE MATEMÁTICA

INSTITUTO SUPERIOR TÉCNICO



UFRGS
INSTITUTO DE INFORMÁTICA
BIBLIOTECA

UFRGS INSTITUTO DE INFORMÁTICA BIBLIOTECA			
N.º CHAMAD :		N.º REG :	
FL 2936		35334	
		DATA:	
		27, 12, 95	
ORIGEM:	DATA:	PREÇO	
D	27, 12, 95	R\$ 15,00	
FUNDO:	FORN.:		
II	II		

DEPARTAMENTO DE MATEMÁTICA - INSTITUTO SUPERIOR TÉCNICO
 AVENIDA ROVISCO PAIS - 1096 LISBOA CODEX
 TEL. 351-1- 841 70 00 FAX. 351-1- 849 92 42

UFRGS
INSTITUTO DE INFORMÁTICA
BIBLIOTECA

Teoria da Ciência da Computação
 Teoria: Ciência: Computação SE
 Redes: Petri
 CNPq 1.03.01.00-3

Synchronization in Petri Nets

P. Blauth MENEZES[†] and J. Félix COSTA^{††}

[†] *Departamento de Matemática, Instituto Superior Técnico*

Av. Rovisco Pais, 1096 Lisboa Codex, Portugal - blauth@solo.inesc.pt

^{††} *Departamento de Informática, Faculdade de Ciências, Universidade de Lisboa*

Campo Grande, 1700 Lisboa, Portugal - fgc@di.fc.ul.pt

Abstract. In "Petri nets are Monoids" by Meseguer and Montanari, categories for Petri nets with and without markings are introduced, where the categorical product and coproduct express the joint behavior of nets. However, this framework lacks structure in the sense that there is no categorical technique to define a composition of nets satisfying some given synchronization specification. In this paper, a synchronization operation is proposed which is a functor induced by some given synchronization prescription at the transition level. Moreover, since this operation is also able to represent the asynchronous composition, the fact that some categories of Petri nets lack coproducts (asynchronous composition) is not anymore a restriction for interaction semantics.

1 Introduction

"Petri nets are Monoids" [13] provides a formal basis over graphs and categories for expressing the semantics of concurrent languages in terms of Petri nets. The categories introduced for Petri nets with and without initial markings are equipped with product and coproduct constructions corresponding to parallel composition and nondeterministic choice, respectively. However, this framework lacks structuring in the sense that there is no categorical technique to define a specific composition of nets satisfying some given synchronization prescription. Since complex systems are structured entities, we should be able to reason and built on their parts separately. Moreover, as showed in [18], nets with markings (and asynchronous morphisms) do not have coproducts. A simple solution for this problem is proposed in [13]: the multiplicity of each place in the initial marking can not be greater then one. However, coproduct becomes a kind of "total choice" instead of an asynchronous composition.

The approach proposed in this work introduces a synchronization operation which is a functor induced by a fibration plus a morphism specifying the transitions to be synchronized. The functor applied to the categorical product of nets (all possible combination between transitions) erases all those transitions which do not reflect the desired synchronization. An important result is that this construction is able to represent synchronous and asynchronous compositions and therefore, the existence of the categorical coproduct (asynchronous composition) is not anymore important for interaction semantics. Moreover, no restriction on nets (such as safe nets as in [18]), initial markings (such as sets of initial markings, instead of multisets as in [13]) or morphisms is necessary.

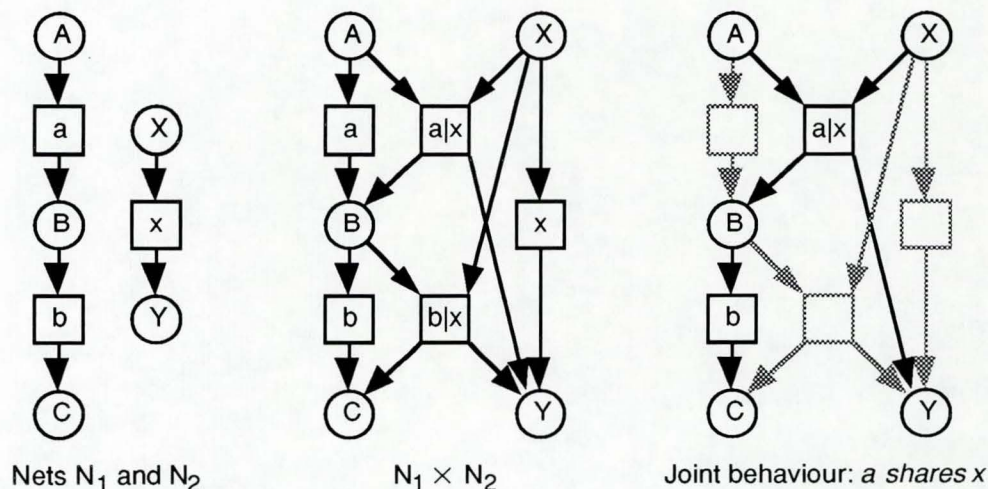
Also, two kinds of synchronizations are categorically defined. The calling is a non-symmetrical relation where a calls x means that the happening of a leads to the synchronous happening of x and the sharing is a symmetrical relation where a shares x is the same as a calls x and x calls a . Note that the synchronization operation proposed is generic and the extension for any other criterion of synchronization is straightforward.

Three categories of Petri nets (from [13]) are introduced:

- $\mathcal{P}etri$ of Petri nets represented as graphs with monoidal structure on states and structure preserving graph morphisms. The states are structured as a free commutative monoid generated by a set of places, where the multiplicity of a place means more than one token produced or consumed simultaneously.
- $\mathcal{P}etri_{\checkmark}$ of pointed Petri nets. It is analogous to $\mathcal{P}etri$ except that each net has a distinguished transition \checkmark , called skip transition, which allow us to "forget" or "erase" some transitions in net morphisms, resulting in a product construction more suitable and useful for synchronization of nets.
- $\mathcal{M}\mathcal{P}etri_{\checkmark}$ of marked (pointed) Petri nets, where each object is a pointed Petri net with an initial marking (any element of the free commutative monoid of states). Since initial markings become part of the structure of the nets, they are preserved by morphisms.

In $\mathcal{P}etri$, the coproduct results in just putting together the nets and can be viewed as an asynchronous composition. The categorical product is the composition operation where the transitions of the resulting net are pairs of transitions from the given nets. Each pair represents the synchronization between component transitions. Thus, the composed net reflects a kind of "total synchronization": each transition of the first net is combined with all transitions of the second net. Therefore, the product construction has very few practical applications, in the sense that usually we want to synchronize some but not all transitions.

The categorical coproduct in $\mathcal{P}etri_{\checkmark}$ has the same interpretation as in $\mathcal{P}etri$. However, the product is very different and the resulting net represents all possible combination between component transitions, with and without synchronization. To obtain the synchronized net, we "erase" from the product all those transitions which do not reflect some given synchronization specification. For instance, consider the figure below:



To obtain the joint behavior of N_1, N_2 where a shares x , we erase from $N_1 \times N_2$ all transitions related to a or x except $a|x$. The categorical technique used is that of a fibration which is also used e.g. in [18]: the joint behavior is obtained applying the functor induced

by a synchronization morphism (which specifies the transitions to be synchronized) and a fibration (a forgetful functor from $\mathcal{Petri}_{\checkmark}$ into the category of pointed sets) to the product of nets. The table of synchronizations (transitions of the synchronized net) for calling and sharing (or both) is categorically defined and the synchronization morphism is uniquely induced.

As stated in [13], Petri nets with initial markings are necessary for defining the operational semantics of concurrent languages (see, for instance, [4], [5], [17], [14] and [6]). The categorical product in $\mathcal{MPetri}_{\checkmark}$ has the same interpretation as in $\mathcal{Petri}_{\checkmark}$ and the synchronization operation is easily extended for marked nets. Therefore, the fact that $\mathcal{MPetri}_{\checkmark}$ lacks all coproducts is not anymore a restriction for interaction semantics of Petri nets.

2 Petri Nets

First we introduce the concept of graph, graph morphism and the corresponding category and then we define Petri nets as graphs leading to a category \mathcal{Petri} . A pointed Petri net is a Petri net where the set of transitions has a distinguished element \checkmark called skip transition. When a morphism maps a transition into \checkmark , it is the same as to forget that transition. Pointed Petri nets and its morphisms constitute the category $\mathcal{Petri}_{\checkmark}$.

2.1 Petri Nets as Graphs

In what follows, suppose that k is in $\{0, 1\}$.

Definition 2.1 Graph. A (small) graph G is a quadruple $\langle V, T, \partial_0, \partial_1 \rangle$ where T is a set of arcs, V is a set of nodes and $\partial_0, \partial_1: T \rightarrow V$ are total functions called source and target, respectively. \square

Definition 2.2 Graph Morphism. A graph morphism $h: G_1 \rightarrow G_2$ where $G_1 = \langle V_1, T_1, \partial_{01}, \partial_{11} \rangle$ and $G_2 = \langle V_2, T_2, \partial_{02}, \partial_{12} \rangle$ is a pair of total functions $h = \langle h_V: V_1 \rightarrow V_2, h_T: T_1 \rightarrow T_2 \rangle$ such that $h_V \circ \partial_{k1} = \partial_{k2} \circ h_T$. \square

Graphs and graph morphisms constitute the category \mathcal{Graph} . A transition t such that $\partial_0(t) = X$ and $\partial_1(t) = Y$ is denoted by $t: X \rightarrow Y$.

A Petri net, in this paper, means the general case of a place/transition net. We introduce the standard definition of a place/transition net as in [15] and then Petri nets as graphs. For further details see [13].

Definition 2.3 Place/Transition Net. A place/transition net is a triple $\langle S, T, F \rangle$ where S is a set of places, T is a set of transitions and $F: (S \times T) + (T \times S) \rightarrow \mathbb{N}$ is the causal dependency relation (F is a multiset and \mathbb{N} is the set of natural numbers). \square

The casual dependency relation specifies how many tokens are consumed or produced in each place when a transition fires. For instance,

$$(A, a) \mapsto 3 \text{ and } (a, B) \mapsto 5 \quad \text{represented by} \quad \textcircled{A} \xrightarrow{3} \boxed{a} \xrightarrow{5} \textcircled{B}$$

specifies that when the transition a fires 3 tokens are consumed at A and 5 tokens are produced at B . For simplicity, in graphical representation, an arc labeled by 1 has its value omitted.

To define a Petri net as a graph, we consider the states as a free commutative monoid generated by a set of places. In this case, with respect to each transition, n tokens consumed or produced at state A is represented by nA and n_i tokens consumed or produced simultaneously at state A_i with i ranging over $1, \dots, p$ respectively, is represented by $n_1A_1 \oplus n_2A_2 \oplus \dots \oplus n_pA_p$, where \oplus is the operation of the free commutative monoid.

In what follows, \mathcal{CMon} denotes the category of free commutative monoids and $\mathcal{cs}: \mathcal{CMon} \rightarrow \mathcal{Set}$ is the canonical forgetful functor.

Definition 2.4 Petri net. A (place/transition) Petri net is a quadruple $N = \langle S^\oplus, T, \partial_0, \partial_1 \rangle$ where S^\oplus is the free commutative monoid generated by a set S and $\partial_0, \partial_1: T \rightarrow \mathcal{cs}S^\oplus$ are total functions. \square

The elements of S, S^\oplus and T are called places, states and transitions, respectively.

Definition 2.5 Petri Net Morphism. A Petri net morphism $h: N_1 \rightarrow N_2$ where $N_1 = \langle S_1^\oplus, T_1, \partial_{01}, \partial_{11} \rangle, N_2 = \langle S_2^\oplus, T_2, \partial_{02}, \partial_{12} \rangle$ is a pair $h = \langle h_S: S_1^\oplus \rightarrow S_2^\oplus, h_T: T_1 \rightarrow T_2 \rangle$ such that h_S is a \mathcal{CMon} -morphism, h_T is a total function and $\mathcal{cs}h_S \circ \partial_{k1} = \partial_{k2} \circ h_T$. \square

Petri nets and its morphisms constitute the category \mathcal{Petri} . The categorical product and coproduct of two nets $N_1 = \langle S_1^\oplus, T_1, \partial_{01}, \partial_{11} \rangle, N_2 = \langle S_2^\oplus, T_2, \partial_{02}, \partial_{12} \rangle$ are as follows (remember that products and coproducts of free commutative monoids are isomorphic):

$$\begin{aligned} N_1 \times_{\mathcal{Petri}} N_2 &= \langle (S_1 +_{\mathcal{Set}} S_2)^\oplus, T_1 \times_{\mathcal{Set}} T_2, \partial_{01} \times_{\mathcal{Set}} \partial_{02}, \partial_{11} \times_{\mathcal{Set}} \partial_{12} \rangle \\ N_1 +_{\mathcal{Petri}} N_2 &= \langle (S_1 +_{\mathcal{Set}} S_2)^\oplus, T_1 +_{\mathcal{Set}} T_2, \partial_{01} +_{\mathcal{Set}} \partial_{02}, \partial_{11} +_{\mathcal{Set}} \partial_{12} \rangle \end{aligned}$$

For simplicity, whenever possible, the identification of the category in products and coproducts is omitted. The functions $\partial_{k1} \times \partial_{k2}$ and $\partial_{k1} + \partial_{k2}$ above are uniquely induced by the product and coproduct constructions, respectively. Intuitively, the product and coproduct constructions in \mathcal{Petri} are viewed as follows:

- product: the composition operation with (total) synchronization in the sense that each transitions of the first net are synchronized with all transitions of the second;
- coproduct: the asynchronous composition operation. It is just the result of putting together the two nets, without any synchronization between component transitions.

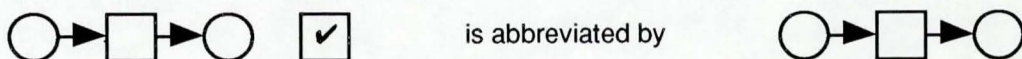
2.2 Pointed Petri Nets

Since a \mathcal{Petri} -morphism $h: N_1 \rightarrow N_2$ is a pair of total functions, each transition of N_1 is mapped onto a transition of N_2 . To forget (erase) some transitions in net morphisms, we consider the set of transitions as a pointed set and require that the transition map is a pointed set homomorphism. A transition mapped onto the distinguished element is "forgotten".

In what follows, \mathcal{Set}^\bullet denotes the category of pointed sets and $\mathcal{cs}_p: \mathcal{CMon} \rightarrow \mathcal{Set}^\bullet$ is the canonical forgetful functor which takes the unity of the monoid into the distinguished element of the corresponding pointed set.

Definition 2.6 Pointed Petri Net. A pointed Petri net is a quadruple $\langle S^\oplus, T, \partial_0, \partial_1 \rangle$ such that T is a pointed set and $\partial_0, \partial_1: T \rightarrow \mathcal{cs}_p S^\oplus$ are \mathcal{Set}^\bullet -morphisms. \square

The distinguished element of T , denoted by \checkmark , is called skip transition. Since ∂_k are \mathcal{Set}^\bullet -morphisms, \checkmark is an isolated transition (no token is consumed or produced). For simplicity, in graphical representation we omit the skip transition. For instance:



Definition 2.7 Pointed Petri Net Morphism. A pointed Petri net morphism $h: N_1 \rightarrow N_2$ where $N_1 = \langle S_1^\oplus, T_1, \partial_{01}, \partial_{11} \rangle$, $N_2 = \langle S_2^\oplus, T_2, \partial_{02}, \partial_{12} \rangle$ is a pair $h = \langle h_S: S_1^\oplus \rightarrow S_2^\oplus, h_T: T_1 \rightarrow T_2 \rangle$ such that h_S is a *CMon*-morphism, h_T is a *Set*^{*}-morphism and $cs_P h_V \circ \partial_{k1} = \partial_{k2} \circ h_T$. \square

Pointed Petri nets and its morphisms constitute a category $\mathcal{Petri}_\checkmark$. The categorical product and coproduct of two pointed nets $N_1 = \langle S_1^\oplus, T_1, \partial_{01}, \partial_{11} \rangle$ and $N_2 = \langle S_2^\oplus, T_2, \partial_{02}, \partial_{12} \rangle$ are as follows:

$$N_1 \times N_2 = \langle (S_1 + S_2)^\oplus, T_1 \times T_2, \partial_{01} \times \partial_{02}, \partial_{11} \times \partial_{12} \rangle$$

$$N_1 + N_2 = \langle (S_1 + S_2)^\oplus, T_1 + T_2, \partial_{01} + \partial_{02}, \partial_{11} + \partial_{12} \rangle$$

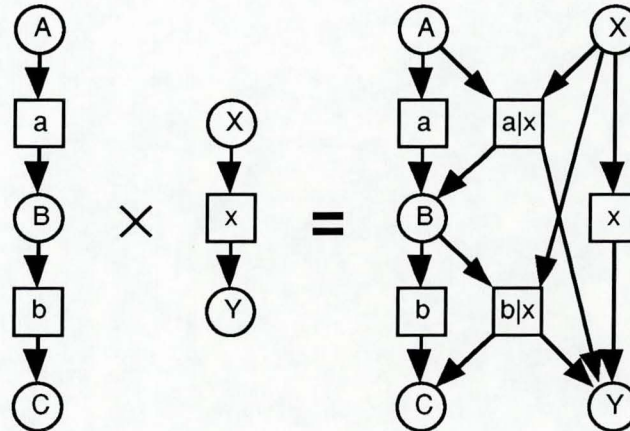
where $\partial_{k1} \times \partial_{k2}$ and $\partial_{k1} + \partial_{k2}$ are uniquely induced by the product and coproduct constructions in *Set*^{*}, respectively. The following notation is used for elements in $T_1 \times T_2$:

$t_1 | t_2$ for (t_1, t_2) meaning the composition with synchronization;

t for (t, \checkmark) or (\checkmark, t) meaning that the transition t is not synchronized.

The coproduct construction in $\mathcal{Petri}_\checkmark$ has the same interpretation as in *Petri*. However, the product in $\mathcal{Petri}_\checkmark$ has a different interpretation and can be viewed as the composition of nets with all possible combinations between transitions, as illustrated in the following example:

Example 2.8 Product in $\mathcal{Petri}_\checkmark$:

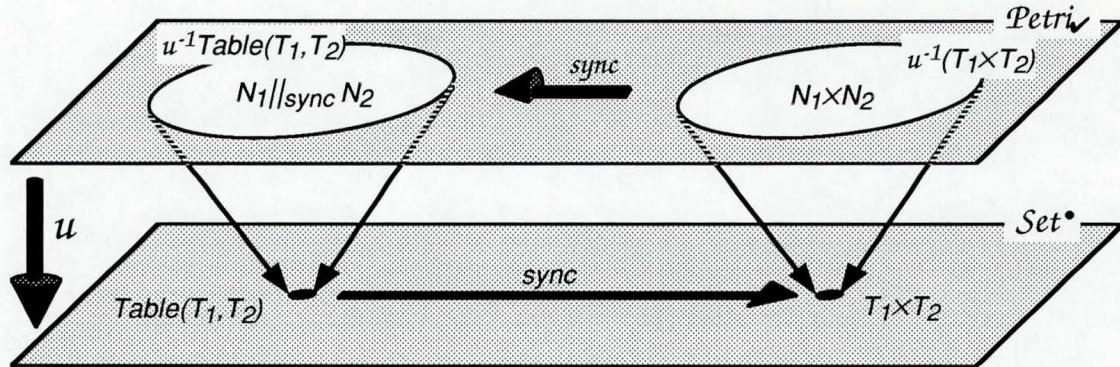


3 Synchronization

The synchronization between nets erases from the product in $\mathcal{Petri}_\checkmark$ all those transitions which do not reflect some given table of synchronizations, as follows (see figure below):

- let $N_1 = \langle S_1^\oplus, T_1, \partial_{01}, \partial_{11} \rangle$, $N_2 = \langle S_2^\oplus, T_2, \partial_{02}, \partial_{12} \rangle$ be pointed Petri nets;
- let $\text{Table}(T_1, T_2)$ be a table of synchronization which contains the pairs of transitions to be synchronized and $\text{sync}: \text{Table}(T_1, T_2) \rightarrow T_1 \times T_2$ be the synchronization morphism which maps the table into the transitions of given nets;
- let $u: \mathcal{Petri}_\checkmark \rightarrow \text{Set}^*$ be the obvious forgetful functor which takes each net into its pointed set of transitions. The functor u is a fibration and the fibers $u^{-1}(\text{Table}(T_1, T_2))$, $u^{-1}(T_1 \times T_2)$ are subcategories of $\mathcal{Petri}_\checkmark$;

d) the fibration u and the morphism $sync$ induce a functor $sync: u^{-1}(T_1 \times T_2) \rightarrow u^{-1}(Table(T_1, T_2))$. The functor $sync$ applied to $N_1 \times N_2$ provides the pointed Petri net reflecting the desired synchronization of the given nets.



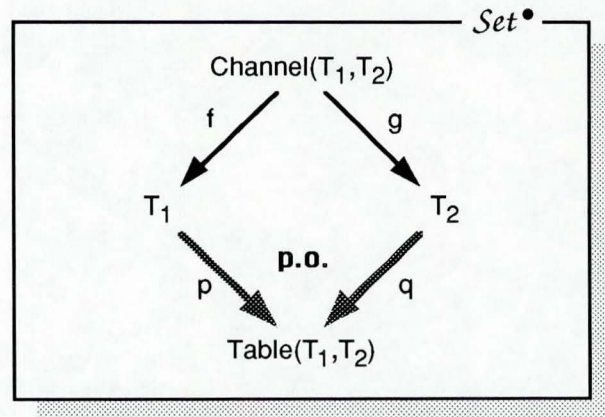
Therefore, the resulting net is determined by the specification of the synchronization at transition level. Note that we are synchronizing transitions and not labels (of transitions) such as in CSP (see [8]). For the synchronization on labels see [12].

In what follows we also show a categorical way to construct the table of synchronization and the corresponding synchronization morphism for sharing and calling.

3.1 Sharing

The table of synchronization for sharing is the resulting object of a pushout whose middle object has as elements pairs of transitions to be synchronized. The corresponding synchronization morphism is uniquely induced by the product construction.

Definition 3.1 Table of Synchronization for Sharing. Let $N_1 = \langle S_1^\oplus, T_1, \partial_{01}, \partial_{11} \rangle$, $N_2 = \langle S_2^\oplus, T_2, \partial_{02}, \partial_{12} \rangle$ be \mathcal{Petri}_\bullet -objects, $Channel(T_1, T_2)$ be the least pointed set which contains all pairs of transitions to be synchronized and $f: Channel(T_1, T_2) \rightarrow T_1$, $g: Channel(T_1, T_2) \rightarrow T_2$ be morphisms which project the components of the pairs into the corresponding pointed set of transitions. The table of synchronization $Table(T_1, T_2)$ is given by the pushout construction as follows:



□

In fact, $Table(T_1, T_2)$ is the disjoint union of T_1 and T_2 except in those elements which are identified by f and g .

Proposition 3.2 Let $\text{Table}(T_1, T_2)$ together with $p: T_1 \rightarrow \text{Table}(T_1, T_2)$, $q: T_2 \rightarrow \text{Table}(T_1, T_2)$ be a pushout which defines the table of synchronization for some given $\text{Channel}(T_1, T_2)$ and $f: \text{Channel}(T_1, T_2) \rightarrow T_1$, $g: \text{Channel}(T_1, T_2) \rightarrow T_2$. Then there are retractions for p and q denoted by p^R and q^R , respectively.

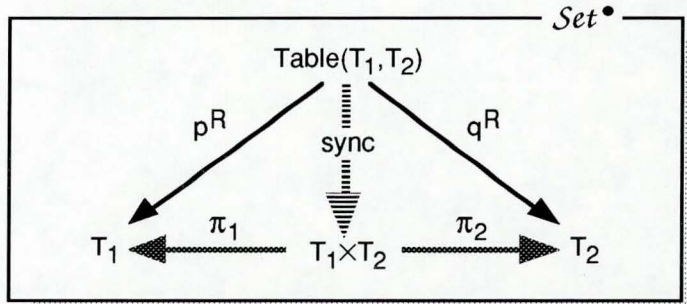
Proof: Since f, g are mono, then p, q are also mono. Thus, there are retractions for p, q and they are defined as follows:

for every b in $\text{Table}(T_1, T_2)$,

if there is a in T_1 such that $p(a) = b$ then $p^R(b) = a$ else $p^R(b) = \checkmark$;

if there is a in T_2 such that $q(a) = b$ then $q^R(b) = a$ else $q^R(b) = \checkmark$. \square

Definition 3.3 Synchronization Morphism. The synchronization morphism $\text{sync}: \text{Table}(T_1, T_2) \rightarrow T_1 \times T_2$ is uniquely induced by the product construction as follows:



\square

Example 3.4 Consider the transition sets $T_1 = \{\checkmark, a, b\}$, $T_2 = \{\checkmark, x, y\}$ and $T_1 \times T_2 = \{\checkmark, a, b, x, y, a|x, a|y, b|x, b|y\}$.

- a shares x , b shares y : $\text{Channel}(T_1, T_2) = \text{Table}(T_1, T_2) = \{\checkmark, a|x, b|y\}$ and $\text{sync}(\checkmark) = \checkmark$, $\text{sync}(a|x) = a|x$, $\text{sync}(b|y) = b|y$;
- no synchronization between transitions: $\text{Channel}(T_1, T_2) = \{\checkmark\}$, $\text{Table}(T_1, T_2) = \{\checkmark, a, b, x, y\}$ and $\text{sync}(\checkmark) = \checkmark$. \square

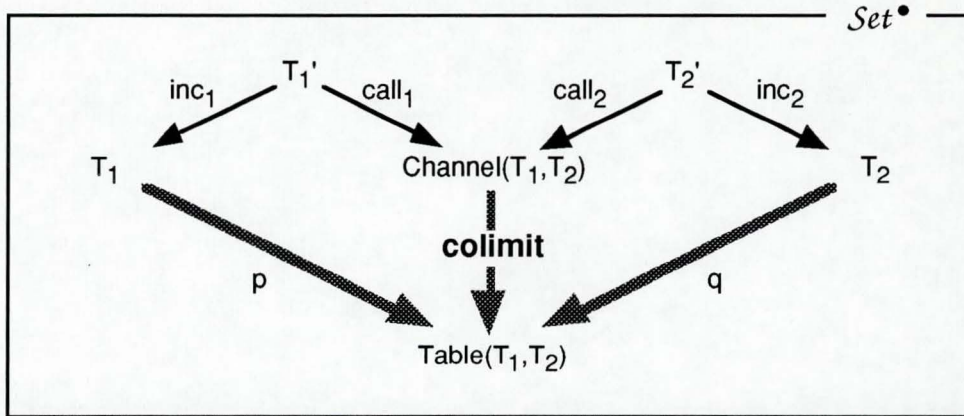
3.2 Calling and Sharing

The table of synchronization for calling and sharing is given by a colimit of a "twin peaks" or "M" diagram (i.e., a diagram with the shape $\bullet \leftarrow \bullet \rightarrow \bullet \leftarrow \bullet \rightarrow \bullet$). Remember that a shares x is the same as a calls x and x calls a .

Definition 3.5 Table of Synchronization. Let $N_1 = \langle S_1^\oplus, T_1, \partial_{01}, \partial_{11} \rangle$, $N_2 = \langle S_2^\oplus, T_2, \partial_{02}, \partial_{12} \rangle$ be $\text{Petri}_{\checkmark}$ -objects and let i be in $\{1, 2\}$:

- let $\text{Channel}(T_1, T_2)$ be the least pointed set which contains all pairs of transitions to be synchronized;
- T_i' is the least pointed subset of T_i containing all transitions of N_i which call a transition of the other net;
- the morphisms $\text{call}_i(a): T_i' \rightarrow \text{Channel}(T_1, T_2)$ are such that:
 - for a in T_i' , if a calls x then $\text{call}_i(a) = a|x$;
 - for a, b in T_i' such that $a \neq b$, a calls x and x calls b is not allowed.

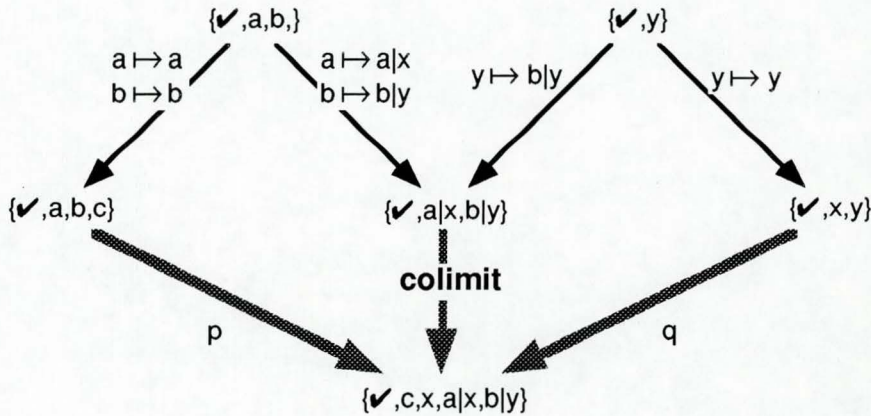
Let $M(T_1, T_2)$ be the twin peaks diagram represented below where $inc_i: T_i' \rightarrow T_i$ are the canonical inclusion morphisms. The table of synchronization $Table(T_1, T_2)$ is given by the colimit of $M(T_1, T_2)$.



□

From the definition above, we can infer that: (from c.1) $call_i$ are monomorphisms and (from c.2) the happening of a transition may not lead to the happening of a different transition in the same net.

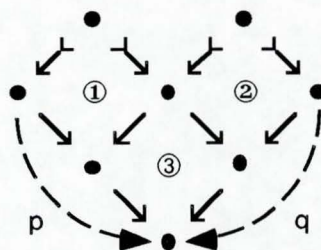
Example 3.6 Consider the transition sets $T_1 = \{\checkmark, a, b, c\}$ and $T_2 = \{\checkmark, x, y\}$. Suppose that a calls x , b calls y and y calls b (i.e., b shares y). Then, $Channel(T_1, T_2) = \{\checkmark, a \mid x, b \mid y\}$ and $Table(T_1, T_2)$ is determined as follows:



□

Proposition 3.7 Consider the diagram $M(T_1, T_2)$ whose colimit determines $Table(T_1, T_2)$ and the morphisms $p: T_1 \rightarrow Table(T_1, T_2)$, $q: T_2 \rightarrow Table(T_1, T_2)$. Then there are retractions for p and q denoted by p^R and q^R , respectively.

Proof: The colimit of $M(T_1, T_2)$ can be determined by pushouts ①, ②, ③ as follows:



Since all morphisms in the above diagram are mono ($\text{inc}_i, \text{call}_i$ are mono and therefore, the morphisms of the pushouts are also mono) and since the composition of monomorphisms is mono, then p, q are mono. Thus, there are retractions for p, q as follows:

for every b in $\text{Table}(T_1, T_2)$,

if there is a in T_1 such that $p(a) = b$ then $p^R(b) = a$ else $p^R(b) = \checkmark$;

if there is a in T_2 such that $q(a) = b$ then $q^R(b) = a$ else $q^R(b) = \checkmark$. \square

Definition 3.8 Synchronization Morphism. The synchronization morphism $\text{sync}: \text{Table}(T_1, T_2) \rightarrow T_1 \times T_2$ is uniquely induced by the product $T_1 \times T_2$ and by the morphisms $p^R: \text{Table}(T_1, T_2) \rightarrow T_1, q^R: \text{Table}(T_1, T_2) \rightarrow T_2$. \square

3.3 Synchronization Functor

First we show that the forgetful functor from $\text{Petri}_{\checkmark}$ into the category of pointed sets is a fibration and then we introduce the synchronization functor.

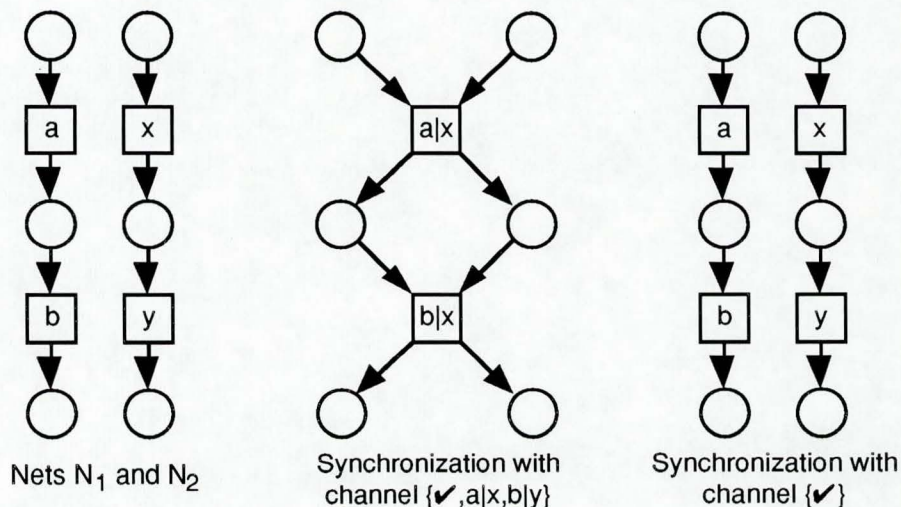
Proposition 3.9 The forgetful functor $u: \text{Petri}_{\checkmark} \rightarrow \text{Set}^\bullet$ which takes each net into its pointed set of transitions is a fibration.

Proof: Let $N_2 = \langle S_2^\oplus, T_2, \partial_{02}, \partial_{12} \rangle$ be a net, $f_T: T_1 \rightarrow T_2$ be a Set^\bullet -morphism and $N_1 = \langle S_1^\oplus, T_1, \partial_{01}, \partial_{11} \rangle$ be a net such that $\partial_{k1} = \partial_{k2} \circ f_T$. Then, $f = \langle \text{id}_{S_2^\oplus}, f_T \rangle: N_1 \rightarrow N_2$ is cartesian with respect to f_T and N_2 . In fact, let $N_3 = \langle S_3^\oplus, T_3, \partial_{03}, \partial_{13} \rangle$ be a net, $v = \langle v_S, v_T \rangle: N_3 \rightarrow N_2$ be a $\text{Petri}_{\checkmark}$ -morphism and $h_T: T_3 \rightarrow T_1$ be a Set^\bullet -morphism such that $v_T = f_T \circ h_T$. Then $h = \langle v_S, h_T \rangle$ is the unique $\text{Petri}_{\checkmark}$ -morphism such that $v = u \circ h$. To see that h is a $\text{Petri}_{\checkmark}$ -morphism consider that $\partial_{k2} \circ v_T = v_S \circ \partial_{k3}$ and, since $v_T = f_T \circ h_T$ and $\partial_{k1} = \partial_{k2} \circ f_T$, we have that $\partial_{k1} \circ h_T = v_S \circ \partial_{k3}$. \square

Definition 3.10 Functor sync. Consider the fibration $u: \text{Petri}_{\checkmark} \rightarrow \text{Set}^\bullet$, the nets $N_1 = \langle S_1^\oplus, T_1, \partial_{01}, \partial_{11} \rangle, N_2 = \langle S_2^\oplus, T_2, \partial_{02}, \partial_{12} \rangle$ and the synchronization morphism $\text{sync}: \text{Table}(T_1, T_2) \rightarrow T_1 \times T_2$. The synchronization of N_1, N_2 represented by $N_1 \parallel_{\text{sync}} N_2$ is given by the functor sync induced by u and sync applied to $N_1 \times N_2$, i.e.:

$$N_1 \parallel_{\text{sync}} N_2 \text{ is } \text{sync}(N_1 \times N_2). \quad \square$$

Example 3.11

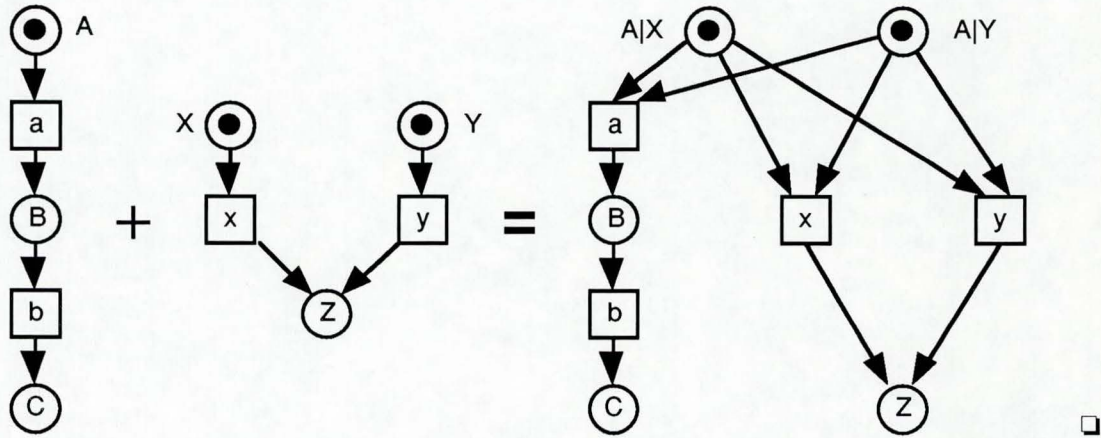


Note that, in the example above, the synchronization with channel $\{\checkmark\}$ results in a net which reflects the asynchronous composition as in the coproduct construction.

4 Marked Petri Nets

As shown in [18], marked nets with asynchronous morphisms do not have coproducts. The solution proposed in [13] restricting initial markings to sets of places (instead of multisets) results in a category with coproducts. But, as illustrated in the example below, the coproduct construction becomes a kind of "total choice" instead of an asynchronous composition of nets as in $\mathcal{Petri}_{\checkmark}$.

Example 4.1 Coproduct of marked Petri nets as in [13]:



Since the categorical product of marked Petri nets (with or without multiplicity in the initial state) has the same interpretation as in $\mathcal{Petri}_{\checkmark}$, we may also introduce the synchronization functor for marked nets. Therefore, the proposed synchronization construction is able to explain the semantics of composed nets with or without synchronization on transitions. Moreover, no restriction on initial markings is needed.

4.1 Petri Nets with an Initial Marking

A Petri net with an initial marking is a pointed Petri net with a distinguished state. The only restriction on morphisms is that initial markings must be preserved. The resulting category has finite products.

Definition 4.2 Marked Petri Net. A marked Petri net is a quintuple $\langle S^{\oplus}, m, T, \partial_0, \partial_1 \rangle$ where $\langle S^{\oplus}, T, \partial_0, \partial_1 \rangle$ is a pointed Petri net and $m = n_1 s_1 \oplus n_2 s_2 \oplus \dots \oplus n_p s_p$ is a distinguished element of S^{\oplus} , called initial state or initial marking. \square

Definition 4.3 Marked Petri Net Morphism. A marked Petri net morphism $h: N_1 \rightarrow N_2$ where $N_1 = \langle S_1^{\oplus}, m_1, T_1, \partial_{01}, \partial_{11} \rangle$, $N_2 = \langle S_2^{\oplus}, m_2, T_2, \partial_{02}, \partial_{12} \rangle$ is a pointed Petri net morphism $h = \langle h_S, h_T \rangle$ such that h_S preserves the initial state, i.e. $h_S(m_1) = m_2$. \square

Marked Petri nets and its morphisms constitute the category $\mathcal{MPetri}_{\checkmark}$. The categorical product of two marked nets $N_1 = \langle S_1^{\oplus}, m_1, T_1, \partial_{01}, \partial_{11} \rangle$, $N_2 = \langle S_2^{\oplus}, m_2, T_2, \partial_{02}, \partial_{12} \rangle$ is as follows:

$$N_1 \times N_2 = \langle (S_1 + S_2)^{\oplus}, m_1 \oplus m_2, T_1 \times T_2, \partial_{01} \times \partial_{02}, \partial_{11} \times \partial_{12} \rangle.$$

4.2 Synchronization

The synchronization in $\mathcal{MPetri}_{\checkmark}$ is defined in the same way as in $\mathcal{Petri}_{\checkmark}$: a functor induced by a fibration and a synchronization morphism. Also, the constructions for calling and sharing are analogous.

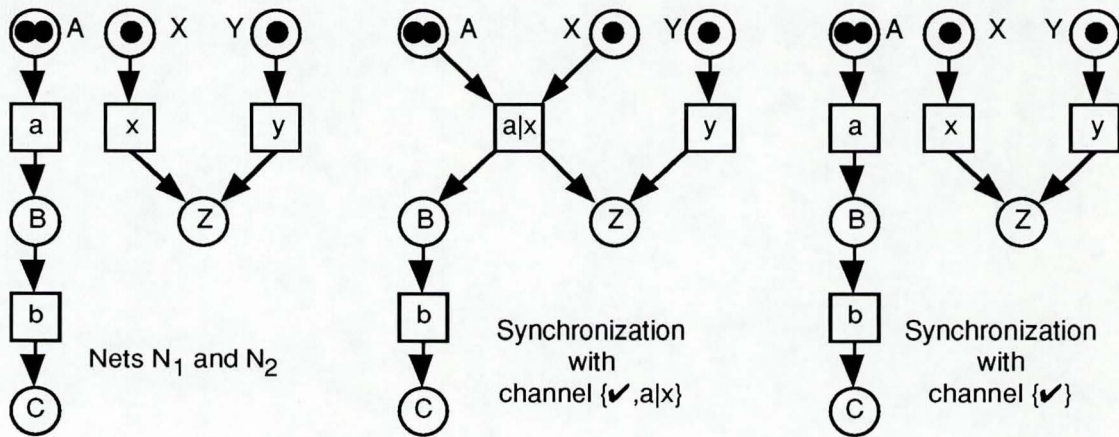
Proposition 4.4 The forgetful functor $mu: \mathcal{MPetri}_{\checkmark} \rightarrow Set^{\bullet}$ which takes each marked net into its pointed set of transitions is a fibration.

Proof: The proof is analogous to the one for $\mathcal{Petri}_{\checkmark}$. □

Definition 4.5 Functor sync. Consider the fibration $mu: \mathcal{MPetri}_{\checkmark} \rightarrow Set^{\bullet}$, the nets $N_1 = \langle S_1^{\oplus}, m_1, T_1, \partial_{01}, \partial_{11} \rangle$, $N_2 = \langle S_2^{\oplus}, m_2, T_2, \partial_{02}, \partial_{12} \rangle$ and the synchronization morphism $sync: Table(T_1, T_2) \rightarrow T_1 \times T_2$. The synchronization of N_1, N_2 represented by $N_1 \parallel_{sync} N_2$ is given by the functor $sync$ induced by mu and $sync$ applied to $N_1 \times N_2$, i.e.:

$$N_1 \parallel_{sync} N_2 \text{ is } sync(N_1 \times N_2). \quad \square$$

Example 4.6 Compare the synchronization with channel $\{\checkmark\}$ below with the example 4.1:



5 Concluding Remarks

In the context of "Petri Nets are Monoids" [13], we solved the problem of how to categorically explain the composition of nets satisfying some given synchronization prescription. The proposed approach defines a categorical structuring technique called synchronization construction which is a functor induced by a fibration and a synchronization morphism which specifies the transitions to be synchronized. The functor applied to the categorical product of nets (which represents all possible combination between transitions) erases all those transitions which do not reflect the desired synchronization. An important result is that this construction is able to represent synchronous and asynchronous compositions. Therefore, the fact that some categories of Petri nets lack coproduct (asynchronous composition) is not anymore a restriction for interaction semantics. Moreover, no restriction on nets, initial markings or morphisms is necessary.

Also, two kinds of synchronization between transitions are introduced: calling (non-symmetrical relation) and sharing (symmetrical relation). The table of synchronization (transitions of the synchronized net) for calling, sharing or both is also categorically defined and the synchronization morphism is uniquely determined.

We are generalizing this framework, including the diagonal compositionality requirement i.e., both vertical (compositional refinement of systems) and horizontal (refinement of systems distributes through interacting combinators). Therefore, we should be able to further define levels of abstractions of systems before or after a synchronization composition in order to obtain the same resulting system. We already achieved some results w.r.t. the implementation (generalization of the procedure call for concurrent systems) in [11] where transitions are mapped into transactions and w.r.t. the transformation (generalization of the macro expansion for concurrent systems) in [10] where graph transformations stand for refinements using the so called single pushout approach [9].

Acknowledgments

This work was partially supported by: UFRGS - Universidade Federal do Rio Grande do Sul and CNPq - Conselho Nacional de Desenvolvimento Científico e Tecnológico in Brazil; CEC under ESPRIT-III BRA WG 6071 IS-CORE and BRA WG 6112 COMPASS; ESDI under research contract OBLOG; HCM Scientific Network MEDICIS in Portugal.

References

- [1] J. Adámek, H. Herrlich, & G. Strecker, *Abstract and Concrete Categories*, Wiley, 1990.
- [2] M. Barr & C. Wells, *Category Theory for Computing Science*, Prentice Hall, 1990.
- [3] M. A. Bednarczyk, *Categories of Asynchronous Systems*, Ph.D. thesis, technical report 1/88, University of Sussex, 1988.
- [4] P. Degano, R. De Nicola & U. Montanari, *A Distributed Operational Semantics for CCS Based on Condition/Event Systems*, Acta Informatica 26, pp. 59-91, 1988.
- [5] P. Degano & U. Montanari, *Concurrent Histories: A Basis for Observing Distributed Systems* J. Comput. System Sci. 34, Nos. 2/3, pp. 422-462, 1987.
- [6] R. J. van Glabbeek & F. W. Vaandrager, *Petri Net Model for Algebraic Theories of Concurrency*, LNCS 259, pp. 224-242, Springer-Verlag, 1987.
- [7] R. Gorrieri, *Refinement, Atomicity and Transactions for Process Description Language*, Ph.D. Thesis, Università di Pisa, 1990.
- [8] C. A. R. Hoare, *Communicating Sequential Processes*, Prentice Hall, 1985.
- [9] M. Löwe, *Algebraic Approach to Graph Transformation Based on a Single Pushout Derivation with Partial Morphisms*, Techn. Report 90/5, Technische Universität Berlin, 1990.
- [10] P. B. Menezes, *Compositional Reification of Petri Nets*, technical report INESC/RT/74-94, preprint IST/DM/25-94, INESC/IST, Lisbon, 1994. Submitted.
- [11] P. B. Menezes, *Compositional Reification of Concurrent Systems*, technical report INESC/RT/75-94, preprint IST/DM/26-94, INESC/IST, Lisbon, 1994. Submitted.
- [12] P. B. Menezes & J. F. Costa, *Algebraic Implementation of Transition Systems*, technical report INESC/RT/72-93, preprint IST/DM/3-94, INESC/IST, Lisbon, 1993. Accepted for presentation in EUROCAST'95.
- [13] J. Meseguer & U. Montanari, *Petri Nets are Monoids*, Information and Computation 88, pp. 105-155, Academic Press, 1990.
- [14] E. R. Olderog, *Operational Petri Net Semantics for CSP* Advances in Petri Nets 1987 (G. Rozenberg, Ed), 196-223, Springer-Verlag, 1987.
- [15] W. Reisig, *Petri Nets: An Introduction*, EATCS Monographs on Theoretical Computer Science 4, Springer-Verlag, 1985.
- [16] V. Sassone, M. Nielsen & G. Winskel, *A Classification of Models for Concurrency*, CONCUR 93: 4th International Conference of Concurrency (E. Best, Ed.), Springer-Verlag, 1993.
- [17] G. Winskel, *Categories of Models for Concurrency*, LNCS 197, Seminar on Concurrency (S. Brookes, Roscoe, G. Winskel, Eds.), pp. 246-267, Springer-Verlag, 1984.
- [18] G. Winskel, *Petri Nets, Algebras, Morphisms and Compositionality*, Information and Computation 72, pp. 197-238, Academic Press, 1987.