UNIVERSIDADE TÉCNICA DE LISBOA
IST

DEPARTAMENTO
DE MATEMÁTICA

INSTITUTO SUPERIOR TÉCNICO

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

INESC/RT/74-94 & IST/DM/25-94

Compositional Reification of Petri Nets *

P. Blauth Menezes†

Abstract. A categorical semantic domain is constructed for the reification of Petri nets based on graph transformations. First, the graph transformation concept (based on the single pushout approach) is extended for Petri nets viewed as graphs with partial morphisms. Classes of transformations stand for reifications where part of a net (usually a transition) is replaced by another (possible complex) net allowing a hierarchical specification methodology. The composition of reifications (i.e., composition of pushouts) is defined, leading to a category of nets and reifications which is complete and cocomplete. Since the reification operation composes, the vertical compositionality requirement of Petri nets is achieved. Then, it is proven that the reification also satisfies the horizontal compositionality requirement, i.e., the reification of nets distributes through parallel composition. Techniques for specification of nets, top down design of nets and a notion of bisimulation between unreified and reified net are provided.

Keywords. Petri nets, net-based semantics, reification, vertical and horizontal compositionality, graph transformation, partial morphisms, category theory.

1 Introduction

Petri nets are one of the first models for concurrency developed and are widely used in many applications. Recently, frameworks based on Petri nets have been proposed for expressing the semantics of concurrent systems in the so-called true concurrency approach as in [Meseguer and Montanari 90], [Winskel and Nielsen 94] and [Brown *et al* 91]. However, Petri nets until now lack of a basic property that any mathematical theory of concurrency should satisfy: we call this property *diagonal compositionality* (or modularity) which is both:

a) *Vertical Compositionality* - As stated by [Gorrieri 90] vertical modularity means a hierarchical specification methodology which allows to add or abstract structure into a concurrent system in different levels of abstraction such as in the top-down or bottom-up design of sequential system. In these cases, a system described at a higher level abstracts some details which are further described as (possible) complex entities at lower level. Moreover, several levels of abstraction my be defined in a compositional way. The vertical operator may be of two kinds:
   - *implementation:* a reference for a further definition of an abstraction such as a morphism that maps transitions into transactions. Note that the refinement morphism as proposed in [Winskel and Nielsen 94] and [Brown *et al* 91] which is a mapping of transitions into transitions meaning a (direct) simulation without further detailing is a special case of implementation.
   - *derivation:* replaces a part of the system by another system. It can be viewed as the generalization of the macro expansion for concurrent systems.

b) *Horizontal Compositionality* - Complex systems are structured entities and can be better understood if we can reason and build on their parts separately. In a concurrent system the parallel composition is the main combinator for constructing new processes. However, we should be able to specify the changes of levels of abstractions of a concurrent system (vertical modularity) before or after the joint behaviour of component parts in order to obtain the same resulting system. Thus, the vertical composition should distribute through the parallel combinator.

Our goal is to achieve the diagonal compositionality of Petri nets without adding extra structures neither modifying its basic definition. In this framework we deal with the derivation operation (for implementation see [Menezes 93]). The derivation operation we define is based on graph transformation using the so-called single pushout approach ([Löwe 90] and [Löwe and Ehrig 90]) on a category of graphs with partial morphisms. Graph transformations standing for a hierarchical specification methodology is, for our knowledge, a new approach.

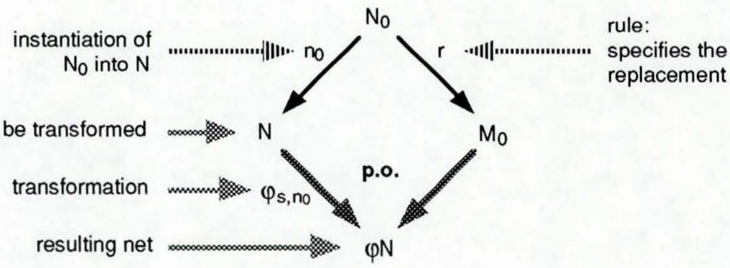
This paper extends the previous work [Menezes 94] mainly with the initial markings for Petri nets and with the notion of simulation between the original net and the derived one. First, we introduce the category of partial Petri nets

* 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 (Information Systems - COrrectness and REusability) and BRA WG 6112 COMPASS (COM-Prehensive Algebraic approach to System Specification and development); ESDI under research contract OBLOG (OBject LOGic) in Portugal.

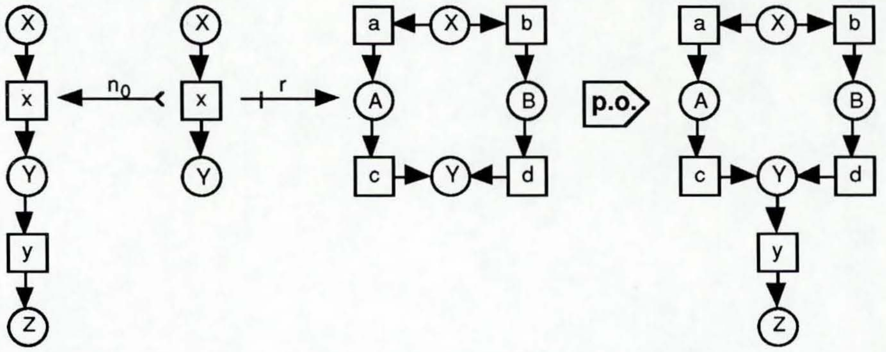
† INESC, Apartado 10105, Lisbon Codex, Portugal; IST, Lisbon, Portugal; E-mail: blauth@inesc.pt.

and partial morphisms which we prove is finitely complete and cocomplete. The category defined follows some ideas introduced in [Meseguer and Montanari 90] and we claim that, with respect to the partial morphisms, "Petri nets are semi-groups". Then, we extend Petri nets with initial markings resulting in a category which is also finitely complete and cocomplete. This is an important result, if we compare, for instance, with [Meseguer and Montanari 90] or [Winskel 87], where the proposed categories are restricted in order the have coproducts. Moreover, it is a basic result for this work, since the reification proposed is based on pushouts.

The graph transformation concept is extended for partial Petri nets with initial markings as follows: a rule $r: N_0 \nrightarrow M_0$ is a partial net morphism which specifies how the net N_0 is replaced by the net M_0 and an instantiation $n_0: N_0 \nrightarrow N$ is a partial net morphism which specifies how N_0 (the source of the rule r) is instantiated into N (the net to be transformed). Then, the transformation φ_{r,n_0} applied to the net N resulting in the net M is given by the pushout construction of r along with n_0, illustrated as follows:

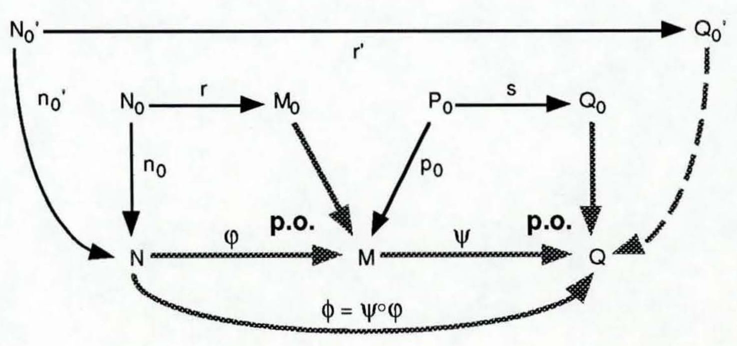


For instance, in the figure below we show how a transition of a net is replaced by a net preserving its source and target nodes. Consider the rule r and the instantiation n_0 where the nodes X and Y are preserved in both r and n_0 and so, they are preserved in the resulting net. However, while the arc x is "forgotten" by r (the morphisms are partial) it is preserved by n_0. Thus, in the resulting net, x is replaced by a, b, c, d and the nodes A and B are added:



Moreover, the transformation operation defined can be used to specify not only further detailing of nets but also some other operations such as *abstraction* which substitutes a (possible) complex part of a net by a simpler net, as in the bottom up design of systems, *deletion* of part of a net or *addition* of new parts to an existing net.

To achieve the vertical composition, we need to compose transformations, i.e., given $\varphi N = M$ and $\psi M = Q$, we should be able to define $\varphi \circ \psi$ such that $(\varphi \circ \psi)N = Q$. Note that the matter is not only the composition of φ and ψ as partial net morphisms, but also as pushouts: given two pushouts with only one vertex in common, we should determine a single pushout such that the resulting transformation is the composition of the component transformations, illustrated as follows:



In fact, we show that for some given rule r and instantiation n_0, the resulting transformation φ_{r,n_0} can be determined by several pushouts. Also, the above composition can be determined by several rules and instantiations. Thus, we can define classes of equivalencies of pairs of rules and instantiations such that the resulting transformation coincide. Therefore, a category of nets and partial morphisms leads to a category where objects are nets and morphisms are classes of equivalence called *reifications*. An important result is that both categories are isomorphic. Then, we show that the *diagonal compositionality* for the category of Petri nets and reifications is achieved.

However, for further specification of a given system, only some reifications are desired. For this purpose, we introduce the definition of grammar which is basically a collection of rules and instantiations to be applied to an initial net. Then, we show how to obtain a subcategory where objects are all nets that can be derived from the initial one and morphisms are all possible reifications (determined by the grammar) to achieve further specifications.

We generalize the top down approach, where rules and instantiations are such that the composition of reifications can be classified as *independent* which means that the second reification derives parts which were *not* previously derived by the first one (it is the sequential independent case of Lowe 90) or *total dependent* which means that the second reification derives parts which were all previously derived by the firs one. The solution proposed for the total dependent case is, for our knowledge, new.

Even if we consider the generalization of the top down approach above, it is possible to reify a Petri net in such way that we can not define a reasonable notion of equivalence between the unreified and the reified nets. Moreover, since a reification may introduce or delete transitions or places, we should not expect a direct simulation between nets. In order to avoid this problem, we introduce the black box reification which replaces single transitions by (possible complex) nets preserving source and target places, among some other constrains. If we consider the part introduced as a "black box" where, except the inherited places, all component transitions and places are hidden, then we are be able to define a notion of bisimulation (which should not be confused with the bisimulation as in [Milner 89]).

2 Partial Petri Nets

First we define partial morphisms on a given category $\mathcal{C}$. Then, we introduce the concepts of graph as an element of a comma category over the base category $\mathcal{S}et$, internal graph which is a graph where the base category is an arbitrary category $\mathcal{C}$ and structured graph which is an extension of the notion of internal graph where arcs and nodes may be objects of different categories, provided that there are functors from these categories to the base category. In this context, the category of Petri nets is defined as the category of partial morphisms on a category of structured graphs. Also, the category of Petri nets with initial markings is introduced. Both categories of Petri nets are finitely complete and cocomplete.

2.1 Categories with Partial Morphisms

For a given category $\mathcal{C}$ we define the partial morphisms in $\mathcal{C}$. If $\mathcal{C}$ has all pullbacks, we can define the composition of partial morphisms leading to the category $p\mathcal{C}$. The main reference for partial morphisms is [Asperti and Longo 91].

Definition 2.1 Partial Morphism. Consider a category $\mathcal{C}$. A partial morphism on $\mathcal{C}$ is an equivalence class of pairs of morphisms $\langle m: D_f \rightarrowtail A, f: D_f \to B\rangle$, where m is mono, with respect to the relation $\langle m: D_f \rightarrowtail A, f: D_f \to B\rangle$ parc $\langle m': D_{f'} \rightarrowtail A, f': D_{f'} \to B\rangle$ if and only if there is an isomorphism $iso: D_f \to D_{f'}$ such that the following diagram commutes:

$$\mathcal{C}$$

$$\begin{tikzcd}
 & D_f \arrow[dl, tail, "m"'] \arrow[dr, "f"] \arrow[dd, dotted, "iso"] & \\
A & & B \\
 & D_{f'} \arrow[ul, tail, "m'"] \arrow[ur, "f'"'] &
\end{tikzcd}$$

❑

Every $\mathcal{C}$-morphism $f: A \to B$ may be represented as a partial morphism $[\langle id_A: A \to A, f: A \to B\rangle]: A \to B$ where id_A is the identity morphism on A. Consider a partial morphism $[\langle m, f\rangle]: A \to B$ where $\langle m: D_f \rightarrowtail A, f: D_f \to B\rangle$ is a representative element of the class. Then $[\langle m, f\rangle]$ is also denoted by $f: A \nrightarrow B$, $\langle m, f\rangle: A \to B$ or $f: A \leftarrowtail D_f \to B$.

Definition 2.2 Category with Partial Morphisms. Consider a category $\mathcal{C} = \langle Ob_{\mathcal{C}}, Mor_{\mathcal{C}}, \partial_0, \partial_1, \iota, \circ\rangle$ with all pullbacks. The category of partial morphism on $\mathcal{C}$ is $p\mathcal{C} = \langle Ob_{\mathcal{C}}, pMor_{\mathcal{C}}, p\partial_0, p\partial_1, \iota, p\circ\rangle$ where $pMor_{\mathcal{C}}, p\partial_0, p\partial_1$

are determined by the definition of partial morphisms on $\mathcal{C}$ and the composition of two morphisms $f = \langle m_f, f\rangle: A \rightarrow B$, $g = \langle m_g, g\rangle: B \rightarrow C$ is $g \circ f = \langle m_g \circ m_f, g \circ f\rangle: A \rightarrow C$ determined by the pullback in the following commutative diagram:

$$\begin{tikzcd}
A & D_f \arrow[l, tail, "m_f"'] \arrow[r, "f"] & B & D_g \arrow[l, tail, "m_g"'] \arrow[r, "g"] & C \\
& & D_f \times_B D_g \arrow[ul] \arrow[ur] \arrow[ull, dashed, tail, bend left, "m_g \circ m_f"] \arrow[urr, dashed, bend right, "g \circ f"'] \arrow[u, phantom, "\text{p.b.}"] & &
\end{tikzcd}$$

$\mathcal{C}$ ❑

For instance, consider the category $\mathcal{S}et$. Then $p\mathcal{S}et$ is the category of sets and partial functions (see [Asperti and Longo 91]). The next proposition shows how a square diagram commutes in $p\mathcal{C}$.

Proposition 2.3 Consider the category $p\mathcal{C}$ and the partial morphisms $f: A \leftarrowtail D_f \rightarrow B$, $g: B \leftarrowtail D_g \rightarrow E$, $u: A \leftarrowtail D_u \rightarrow C$, $v: C \leftarrowtail D_v \rightarrow E$ such that $g \circ f = v \circ u$. Then, there are morphisms $p: D_f \leftarrowtail M \rightarrow D_v$, $q: D_u \leftarrowtail M \rightarrow D_g$ where the middle object M is unique (up to an isomorphism) and are such that the diagram below commutes. Moreover, ① and ② are pullback.

$$\begin{tikzcd}
A & D_f \arrow[l, tail, "m_f"] \arrow[r, "f"] & B \\
D_u \arrow[u, tail, "m_u"] \arrow[d, "u"'] & M \arrow[l, dotted, tail, "m_q"] \arrow[r, dotted, "q"'] \arrow[u, dotted, tail, "m_p"] \arrow[d, dotted, "p"] \arrow[ur, phantom, "①"] \arrow[dl, phantom, "②"] & D_g \arrow[u, tail, "m_g"'] \arrow[d, "g"] \\
C & D_v \arrow[l, tail, "m_v"'] \arrow[r, "v"] & E
\end{tikzcd}$$

$\mathcal{C}$

Proof: The compositions $g \circ f$ and $v \circ u$ are given by pullbacks in ① and ② where $D_f \times_B D_g$ and $D_u \times_C D_v$ are the pullback objects. Since $g \circ f = v \circ u$, there is an isomorphism $iso: D_f \times_B D_g \rightarrow D_u \times_C D_v$ and so, both objects represent the middle object M. ❑

2.2 Graphs

Traditionally, a graph is defined as a quadruple $\langle V, T, \partial_0, \partial_1\rangle$ where V is a set of nodes, T is a set of arcs, and ∂_0, $\partial_1: T \rightarrow V$ are functions called source and target, respectively. However, we prefer a different but equivalent approach which is to consider a graph as an element of a comma category. This approach is used to define graphs, internal graphs and structured graphs. First we introduce the definition of the diagonal functor as in [Mac Lane 71].

Definition 2.4 Diagonal Functor. Consider the category $\mathcal{C}$. Let $\mathcal{C}^2$ be the category where objects and morphism are pairs of objects and morphisms of $\mathcal{C}$. The diagonal functor $\Delta: \mathcal{C} \rightarrow \mathcal{C}^2$ takes each object A into $\langle A, A\rangle$ and each morphism $f: A \rightarrow B$ into $\langle f, f\rangle: \langle A, A\rangle \rightarrow \langle B, B\rangle$. ❑

Definition 2.5 Graph. Consider the diagonal functor $\Delta: \mathcal{S}et \rightarrow \mathcal{S}et^2$. The category of (small) graphs is the comma category $\Delta\downarrow\Delta$ denoted by $\mathcal{G}raph$. ❑

Thus, a graph is a triple $G = \langle V, T, \partial\rangle$ where $\partial = \langle \partial_0, \partial_1\rangle$. We may denote a graph in the traditional way, i.e., $G = \langle V, T, \partial_0, \partial_1\rangle$. As expected, a morphism in $\mathcal{G}raph$ preserves source and target nodes of transitions. It is usual to write $t: X \rightarrow Y$ to denote $\partial_0(t) = X$ and $\partial_1(t) = Y$ for any t in T.

As stated in [Corradini 90] (see also [Asperti and Longo 91] for further details), a (small) graph $G = \langle V, T, \partial_0, \partial_1\rangle$ can be considered as a diagram in the category $\mathcal{S}et$ where V and T are sets and ∂_0, ∂_1 are total functions. Moreover, graph morphisms are commutative diagrams in $\mathcal{S}et$. This means that $\mathcal{S}et$ plays the role of "universe of discourse" of the category $\mathcal{G}raph$: it is defined internally to the category $\mathcal{S}et$. This suggests a generalization of graphs as diagrams in an arbitrary universe category. This approach is known as internalization.

Definition 2.6 Internal Graph. Consider the (base) category $\mathcal{C}$ and the diagonal functor $\Delta: \mathcal{C} \to \mathcal{C}^2$. The category of internal graphs over $\mathcal{C}$ is the comma category $\Delta \downarrow \Delta$, denoted by $\mathcal{G}raph(\mathcal{C})$. ❑

Structured graphs allows the definition of a special kind of graphs where nodes and arcs are object of different categories. They are defined over internal graphs provided that there are functors from the categories of nodes and arcs to the base category. The source and target morphisms are taken from the base category.

Definition 2.7 Structured Graph. Consider the functors $v: \mathcal{V} \to \mathcal{C}$, $t: \mathcal{T} \to \mathcal{C}$ and $\Delta: \mathcal{C} \to \mathcal{C}^2$. The category of structured graphs over the base category $\mathcal{C}$ with respect to the functors v and t denoted by $\mathcal{G}raph(v, t)$ is the comma category $\Delta \circ t \downarrow \Delta \circ v$. ❑

2.3 Petri Nets

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 and then Petri nets viewed as graphs inspired by [Meseguer and Montanari 90].

Definition 2.8 Place/Transition Net. A place/transition net (see for instance [Reisig 85]) is a triple $\langle S, T, F \rangle$ such that S is a set of places, T is a set of transitions and $F: (S \times T) + (T \times S) \to \mathbb{N}$ is the causal dependency relation where F is a multiset (a multiset is a function $f: X \to \mathbb{N}$), $\times$ and $+$ denote the product and the coproduct in $\mathcal{S}et$ and $\mathbb{N}$ denotes the set of natural numbers. ❑

As proposed in [Meseguer and Montanari 90], to represent a Petri net as a graph we can consider the states as elements of a free commutative monoid generated by a set of places. In this case, for each transition, n tokens consumed or produced in the place A is represented by nA and n_i tokens consumed or produced simultaneously in A_i, with i ranging over $1, \dots, k$ is represented by $n_1A_1 \oplus n_2A_2 \oplus \dots \oplus n_kA_k$ (where $\oplus$ is the additive operation of the monoid).

Note that, we may consider that every monoid has a distinguished element which is the unity element. In some sense, the unity element leads to a notion of partiality: to forget an element in a monoid homomorphism it is enough to map this element to the unity of the target object. Considering that we need partial morphism in order to define graph transformations, partial monoid homomorphism can be seen as a partial category of a category which already behaves as a partial one. However, if we consider the category of semi-groups with partial morphisms instead of the category of monoids, the notion of Petri nets as graphs as in [Meseguer and Montanari 90] is kept. Thus, we claim that, for partial morphisms "Petri nets are semi-groups". In what follows, the main reference for concrete categories is [Adámek *et al* 90].

Free Commutative Semi-Groups with Partial Morphisms

The category of free commutative semi-groups with partial morphisms, denoted by $p\mathcal{CS}em$, is concrete over the category of free commutative monoids, denoted by $\mathcal{CM}on$. In fact, any semi-group can be canonically extended as a monoid and a partial semi-group morphism can be viewed as a "pointed" morphism of monoids, where the distinguished element is the unity. Moreover, the limits and colimits of $\mathcal{CM}on$ are lifted to $p\mathcal{CS}em$.

Definition 2.9 Category $p\mathcal{CS}em$. Consider the category of commutative semi-groups, denoted by $\mathcal{CS}em$. The category $p\mathcal{CS}em$ is the category of partial morphisms on $\mathcal{CS}em$. ❑

Proposition 2.10 The category $p\mathcal{CS}em$ is finitely complete and cocomplete.

Proof: Consider the functor $sm: p\mathcal{CS}em \to \mathcal{CM}on$ such that:

- for all commutative semi-group $S^{\oplus}$, $smS^{\oplus} = S_e^{\oplus}$, where $S_e^{\oplus}$ is the free monoid generated by the set S with e as the unity element;
- for all $p\mathcal{CS}em$-morphism $h: S_1^{\oplus} \leftarrow S_h^{\oplus} \to S_2^{\oplus}$, $sm\,h = h_e$ where $h_e: S_{e1}^{\oplus} \to S_{e2}^{\oplus}$ and for all s in $S_1^{\oplus}$, if s is in $S_h^{\oplus}$, then $h_e(s) = h(s)$; else, $h_e(s) = e$.

The functor sm is faithful and so, $\langle p\mathcal{CS}em, sm\rangle$ is a concrete category over $\mathcal{CM}on$. Also, for each finite diagram in $p\mathcal{CS}em$ taken by the functor sm into $\mathcal{CM}on$, the limits and colimits in $\mathcal{CM}on$ can be lifted as an initial source and final sink, respectively, in $p\mathcal{CS}em$. For further details, see [Menezes 94]. ❑

Partial Petri Net

The category of partial Petri nets is defined as the follows:

- first, consider the category of structured graphs where the base category is $p\mathcal{S}et$, the category of arcs is $\mathcal{S}et$ and the category of nodes is $\mathcal{CS}em$ (the category of commutative semi-groups). Thus, the source and target functions are partial.
- then, consider the category of partial morphisms of the above category of structures graphs.

Definition 2.11 Partial Petri net. The category of partial Petri nets is $\mathcal{pPetri} = \mathcal{pGraph}(t, v)$, i.e., the category of partial morphisms on the category of structured graphs $\mathcal{Graph}(t, v)$, where $t\colon \mathcal{Set} \to \mathcal{pSet}$ is the canonical embedding functor and $v\colon \mathcal{CSem} \to \mathcal{pSet}$ is the forgetful functor such that for all $\mathcal{CSem}$-object $S^{\oplus} = \langle S^*, \oplus\rangle$, $vS^{\oplus} = S^*$ and for all $\mathcal{CSem}$-morphism $h\colon S_1^{\oplus} \to S_2^{\oplus}$, $vh\colon S_1^* \to S_2^*$. ❑

Thus, a partial Petri net N is a quadruple $N = \langle V^{\oplus}, T, \partial_0, \partial_0\rangle$ where $V^{\oplus}$ is a free commutative semi-group, T is a set and $\partial_0, \partial_0\colon tT \to vV^{\oplus}$ are partial functions. Let $N_1 = \langle V_1^{\oplus}, T_1, \partial_{01}, \partial_{11}\rangle$ and $N_2 = \langle V_2^{\oplus}, T_2, \partial_{02}, \partial_{12}\rangle$ be nets. From the definition of partial morphism, we infer that a $\mathcal{pPetri}$-morphism $h\colon N_1 \leftarrowtail D_h \to N_2$ is a pair $\langle h_V\colon V_1^{\oplus} \leftarrowtail D_{h_V} \to V_2^{\oplus}, h_T\colon T_1 \leftarrowtail D_{h_T} \to T_2\rangle$ where h_V is a $\mathcal{pCSem}$-morphism, h_T is a partial function and is such that, for k in {0, 1}, the following diagram commutes with ①, ② been pullbacks (m, p are determined by ①, ②, respectively):

$$\begin{tikzcd}
tT_1 & tD_{h_T} \arrow[l, tail, "tm_T"'] \arrow[r, "th_T"] & tT_2 \\
tD_{\partial_{1k}} \arrow[u, tail, "m_{\partial_{1k}}"] \arrow[d, "\partial_{1k}"'] \arrow[ur, phantom, "①"] & tD_{\partial_{hk}} \arrow[l, tail, "tm"] \arrow[r, "tp"'] \arrow[u, tail, "m_{\partial_{hk}}"] \arrow[d, "\partial_{hk}"] \arrow[ur, phantom, "②"] & tD_{\partial_{2k}} \arrow[u, tail, "m_{\partial_{2k}}"'] \arrow[d, "\partial_{2k}"] \\
vV_1^{\oplus} & vD_{h_V} \arrow[l, tail, "vm_V"'] \arrow[r, "vh_V"] & vV_2^{\oplus}
\end{tikzcd}$$

Set

It is easy to prove that, in general, the above diagram is not a commutative diagram in $\mathcal{pSet}$.

Proposition 2.12 The category $\mathcal{pPetri}$ is finitely complete e cocomplete.

<u>Proof:</u> The forgetful functor $v\colon \mathcal{pCSem} \to \mathcal{pSet}$ that takes each semi-group $S^{\oplus} = \langle S^*, \oplus\rangle$ into S^* has left adjoint which takes each set into the commutative semi-group freely generated. Thus, v preserves limits. Suppose k in {0, 1}. For further details in what follows, see [Menezes 94].

a) Zero object. Let 0 and $0^{\oplus}$ be zero objects of $\mathcal{pSet}$ and $\mathcal{pCSem}$, respectively. Then $\langle 0^{\oplus}, 0, !, !\rangle$ where ! is the unique partial function, is a zero object of $\mathcal{pPetri}$.

b) Coproducts. Consider the nets $N_1 = \langle V_1^{\oplus}, T_1, \partial_{10}, \partial_{11}\rangle$ and $N_2 = \langle V_2^{\oplus}, T_2, \partial_{20}, \partial_{21}\rangle$. A coproduct of N_1 and N_2 is the object $N_1+N_2 = \langle V_1^{\oplus}+_{\mathcal{pCSem}}V_2^{\oplus}, T_1+_{\mathcal{pSet}}T_2, \partial_{10}+\partial_{20}, \partial_{11}+\partial_{21}\rangle$ together with the morphisms $q_1 = \langle q_{1V}, q_{1T}\rangle\colon N_1 \to N_1+N_2$ and $q_2 = \langle q_{2V}, q_{2T}\rangle\colon N_2 \to N_1+N_2$ where $\partial_{1k}+\partial_{2k}$ are uniquely induced by the coproduct in $\mathcal{pSet}$, as follows:

$$\begin{tikzcd}
T_1 \arrow[r, "q_{1T}"] \arrow[d, "\partial_{1k}"'] \arrow[dr, dashed] & T_1+T_2 \arrow[d, dashed, "\partial_{1k}+\partial_{2k}"] & T_2 \arrow[l, "q_{2T}"'] \arrow[d, "\partial_{2k}"] \arrow[dl, dashed] \\
v\,V_1^{\oplus} \arrow[r, "v\,q_{1V}"'] & v(V_1^{\oplus}+V_2^{\oplus}) & v\,V_2^{\oplus} \arrow[l, "v\,q_{2V}"]
\end{tikzcd}$$

c) Coequalizers. Consider the nets $N_1 = \langle V_1^{\oplus}, T_1, \partial_{10}, \partial_{11}\rangle$, $N_2 = \langle V_2^{\oplus}, T_2, \partial_{20}, \partial_{21}\rangle$ and the parallel partial morphisms $f, g\colon N_1 \to N_2$ where $f = \langle f_V, f_T\rangle$, $g = \langle g_V, g_T\rangle$. Let $c_V\colon V_2^{\oplus} \to V^{\oplus}$ be a $\mathcal{pCSem}$-coequalizer of f_V, g_V and $c_T\colon T_2 \to T$ be a $\mathcal{pSet}$-coequalizer of f_T, g_T. A coequalizer of f, g is the net $N = \langle V^{\oplus}, T, \partial_0, \partial_1\rangle$ together with the morphism $c = \langle c_V, c_T\rangle\colon N_2 \to N$ where ∂_k are uniquely induced by the coequalizer c_T in $\mathcal{pSet}$, as follows:

$$\begin{tikzcd}
T_1 \arrow[r, shift left, "f_T"] \arrow[r, shift right, "g_T"'] \arrow[d, "\partial_{1k}"'] & T_2 \arrow[r, "c_T"] \arrow[d, "\partial_{2k}"'] \arrow[dr, dashed] & T \arrow[d, dashed, "\partial_k"] \\
v\,V_1^{\oplus} \arrow[r, shift left, "v\,f_V"] \arrow[r, shift right, "v\,g_V"'] & v\,V_2^{\oplus} \arrow[r, "v\,c_V"'] & v\,V_2^{\oplus}
\end{tikzcd}$$

d) Products. Consider the nets $N_1 = \langle V_1{}^{\oplus}, T_1, \partial_{1_0}, \partial_{1_1}\rangle$ and $N_2 = \langle V_2{}^{\oplus}, T_2, \partial_{2_0}, \partial_{2_1}\rangle$. A product of N_1 and N_2 is the object $N_1 \times N_2 = \langle V_1{}^{\oplus} \times_{\mathit{pCSem}} V_2{}^{\oplus}, T_1 \times_{\mathit{pSet}} T_2, \partial_{1_0} \times \partial_{2_0}, \partial_{1_1} \times \partial_{2_1}\rangle$ together with the morphisms $\pi_1 = \langle \pi_{1_V}, \pi_{1_T}\rangle$: $N_1 \times N_2 \to N_1$ and $\pi_2 = \langle \pi_{2_V}, q_{2_T}\rangle$: $N_1 \times N_2 \to N_2$ where $\partial_{1_k} \times \partial_{2_k}$ are uniquely induced by the product in *pCSem,* taken into *pSet,* as follows (remember that v preserves limits):

$$\begin{tikzcd}
T_1 \arrow[d, "\partial_{1_k}"'] & T_1 \times T_2 \arrow[l, "\pi_{1_T}"'] \arrow[r, "\pi_{2_T}"] \arrow[d, dotted, "\partial_{1_k} \times \partial_{2_k}"] \arrow[dl, dashed, bend right] \arrow[dr, dashed, bend left] & T_2 \arrow[d, "\partial_{2_k}"] \\
v\, V_1{}^{\oplus} & v\, V_1{}^{\oplus} \times v\, V_2{}^{\oplus} \arrow[l, "v\,\pi_{1_V}"] \arrow[r, "v\,\pi_{2_V}"'] & v\, V_2{}^{\oplus}
\end{tikzcd}$$

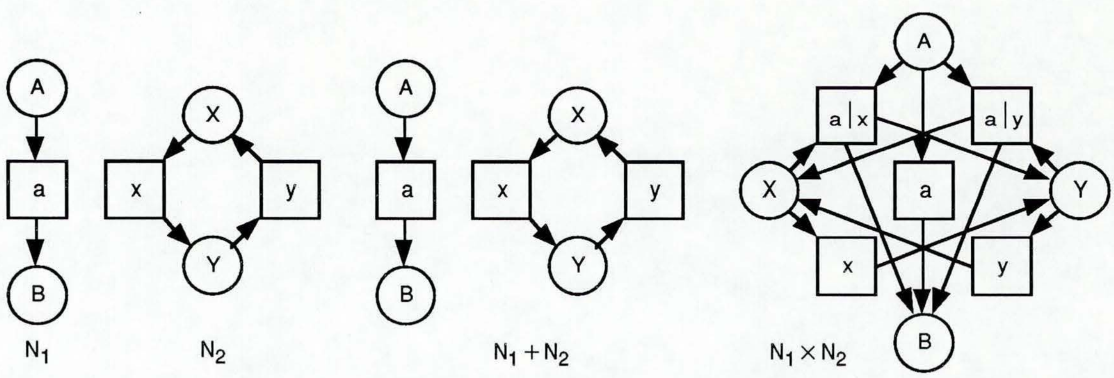
e) Equalizers. Consider the nets $N_1 = \langle V_1{}^{\oplus}, T_1, \partial_{1_0}, \partial_{1_1}\rangle$, $N_2 = \langle V_2{}^{\oplus}, T_2, \partial_{2_0}, \partial_{2_1}\rangle$ and a pair of parallel morphisms f, g: $N_1 \to N_2$ where $f = \langle f_V, f_T\rangle$, $g = \langle g_V, g_T\rangle$. Let e_V: $V^{\oplus} \to V_1{}^{\oplus}$ be a *pCSem*-equalizer of f_V, g_V and e_T: $T \to T_1$ be a *pSet*-equalizer of f_T, g_T. An equalizer of f, g is the net $N = \langle V^{\oplus}, T, \partial_0, \partial_1\rangle$ together with the morphism $e = \langle e_V, e_T\rangle$: $N \to N_1$ where ∂_k are uniquely induced by the equalizer e_V in *pCSem,* taken into *pSet,* as follows (again, remember that v preserves limits):

$$\begin{tikzcd}
T \arrow[r, "e_T"] \arrow[d, dotted, "\partial_k"'] \arrow[dr, dashed, bend left] & T_1 \arrow[r, shift left, "f_T"] \arrow[r, shift right, "g_T"'] \arrow[d, "\partial_{1k}"] & T_2 \arrow[d, "\partial_{2k}"] \\
v\, V_1{}^{\oplus} \arrow[r, "v\, e_V"'] & v\, V_1{}^{\oplus} \arrow[r, shift left, "v\, f_V"] \arrow[r, shift right, "v\, g_V"'] & v\, V_2{}^{\oplus}
\end{tikzcd}$$

❑

In *pPetri,* the coproduct represents the asynchronous composition of nets and the product can be viewed as the parallel composition of nets where all possible combination of component transitions are represented.

Example 2.13 Coproduct and product in *pPetri:*



❑

Remark 2.14 Synchronization of Petri Nets. In our previous work [Menezes and Costa 93], we construct a functorial operation for synchronization of nets, defined for calling and sharing. It is defined using the fibration technique. The synchronization operation erases from the parallel composition (categorical product) of given nets all those transition which do not reflect the given synchronization specification. ❑

2.4 Petri Nets with Initial Marking

A Partial Petri net with initial markings is a partial Petri net endowed with a set of initial markings where the choice of which initial marking is considered at run time is an external nondeterminism. The main advantage of considering a set of initial marking instead of a single initial marking as in [Winskel 87] or [Meseguer & Montanari 90] is that the resulting category has finite colimits. This solution is more general than restricting the category for safe nets as in [Winskel 87] or considering initial marking with one token at most in each place as in [Meseguer & Montanari 90]. Moreover, the coproduct construction reflects the asynchronous composition of component nets.

Definition 2.15 Partial Petri Net with Initial Marking. Consider the category *pPetri.* Let *u: pPetri* →

pSet be a functor such that each *pPetri*-net $N = \langle V^{\oplus}, T, \partial_0, \partial_1\rangle$ with $V^{\oplus} = \langle V^*, \oplus\rangle$ is taken into the set V^* and each *pPetri*-morphism $h = \langle h_V, h_T\rangle$ is taken into the partial function canonically induced by the *pCSem*-morphism h_V. The category of partial Petri nets with initial markings, denoted by *pMPetri*, is the comma category $id_{pSet}\downarrow u$, where id_{pSet} is the identity functor in *pSet*. ❑

Therefore, a partial Petri net with initial markings M is a triple $M = \langle N, I, init\rangle$ where $N = \langle V^{\oplus}, T, \partial_0, \partial_1\rangle$ is a partial Petri net, I is the set of initial states or initial markings and inic is the partial function which instantiates the initial states into the net N. Thus, a net M may also be considered as $M = \langle V^{\oplus}, T, \partial_0, \partial_1, I, init\rangle$. If init is the canonical inclusion, it may be omitted, i.e., $\langle V^{\oplus}, T, \partial_0, \partial_1, I, inclusion\rangle$ is abbreviated by $\langle V^{\oplus}, T, \partial_0, \partial_1, I\rangle$. A *pMPetri*-morphism is a pair $h = \langle h_N, h_I\rangle$. Since h_N is a pair $h_N = \langle h_V, h_T\rangle$, we also represent a *pMPetri*-morphism as a triple $h = \langle h_V, h_T, h_I\rangle$.

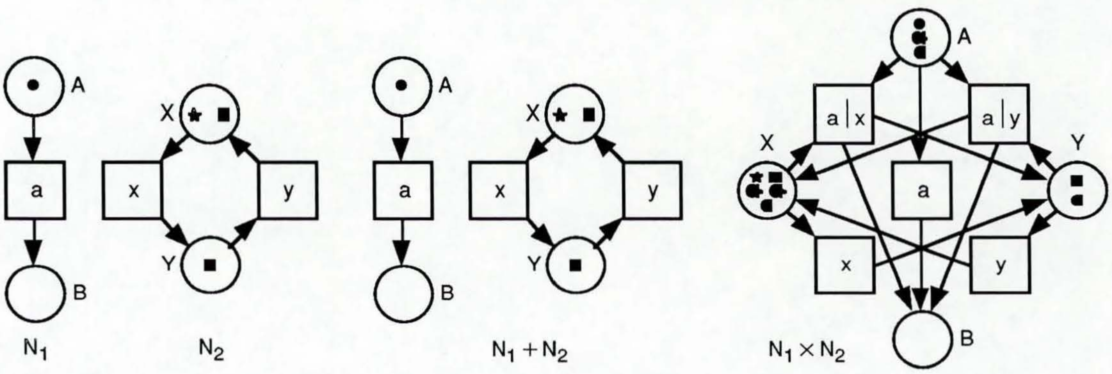
Proposition 2.16 The category *pMPetri* is finitely complete and cocomplete.

Proof: Since *pMPetri* is the comma category $id_{pSet}\downarrow u$, we have only to prove that the functor u: *pPetri* $\to$ *pSet* preserves limits. Consider the initial object { } and the functor p: *pSet* $\to$ *pPetri* such that for all set V, pV is the net $\langle V^{\oplus}, \{\ \}, !, !\rangle$ where $V^{\oplus}$ is the semi-group freely generated from V. The functor p is left adjoint to u. ❑

The product and coproduct in *pMPetri* have the same interpretation as in *MPetri*, i.e., the parallel composition and asynchronous composition, respectively.

Example 2.17 Product and coproduct in *pMPetri*. For the nets represented below, the set of initial markings are the following: $I_1 = \{A\}$, $I_2 = \{X, X+Y\}$, $I_1 + I_2 = \{A, X, X+Y\}$, $I_1 \times I_2 = \{A, X, X+Y, A+X, A+X+Y\}$. The possible initial marking in $I_1 \times I_2$ are represented using the following symbols:

● A ✦ X ■ X+Y ◆ A+X ◀ A+X+Y



❑

3 Reification

The reification defined extends the single pushout approach of graph transformation to Petri nets.

3.1 Transformation of Petri Nets

In what follows, we introduce the concepts of rule, instantiation and transformation.

Definition 3.1 Rule, Instantiation, Transformation.

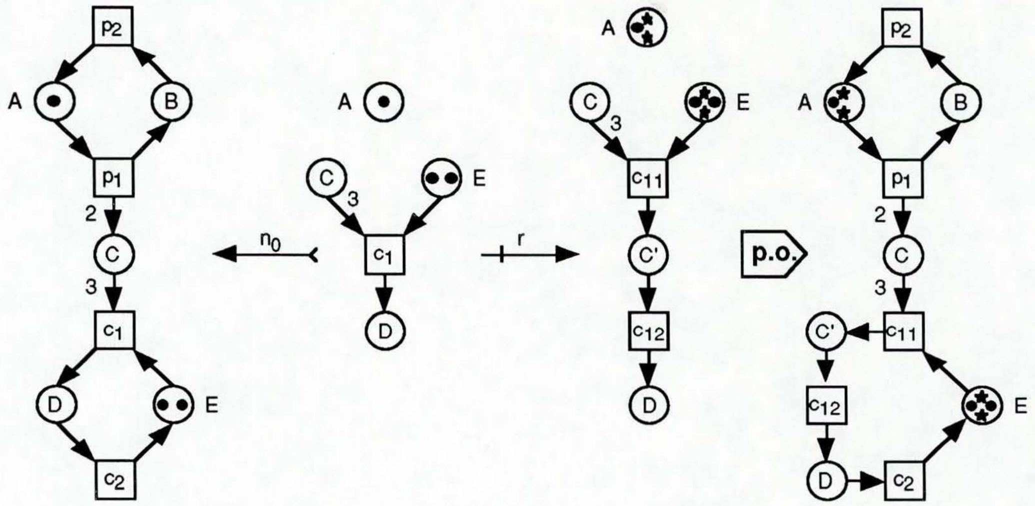
a) A rule $r\colon N_0 \to M_0$ and an instantiation $n_0\colon N_0 \to N$ are just *pMPetri*-morphisms.
b) The transformation of a net N determined by a rule $r\colon N_0 \to M_0$ and an instantiation $n_0\colon N_0 \to M$ is given by the pushout illustrated below, where M is the transformed net and $\varphi_{r,n_0}\colon N \to M$ is the transformation morphism.

pMPetri

$$\begin{tikzcd} & N_0 \arrow[dl, "n_0"'] \arrow[dr, "r"] & \\ N \arrow[dr, "\varphi_{r,no}"'] & \textbf{p.o.} & M_0 \arrow[dl] \\ & M & \end{tikzcd}$$

❑

Example 3.2 Consider the rule r, the instantiation n_0 and the transformed net, as in the figure below. Entities preserved by morphisms are identified with the same label. Note that c_1 is replaced by a sequence of transitions c_{11}, c_{12} and that the state C' is introduced in the resulting net. With respect to the initial markings, the original one is preserved and a second marking is introduced.



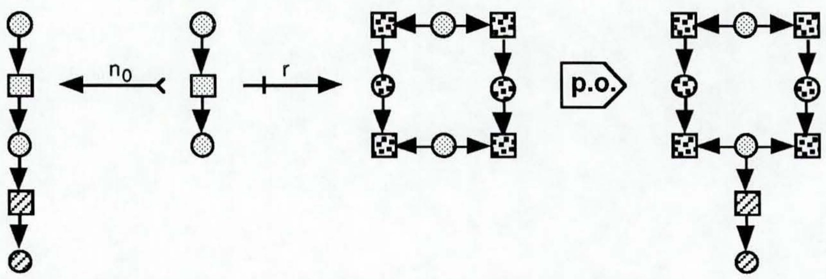
❑

A transformation of a Petri net may be classified in one of the follows cases:

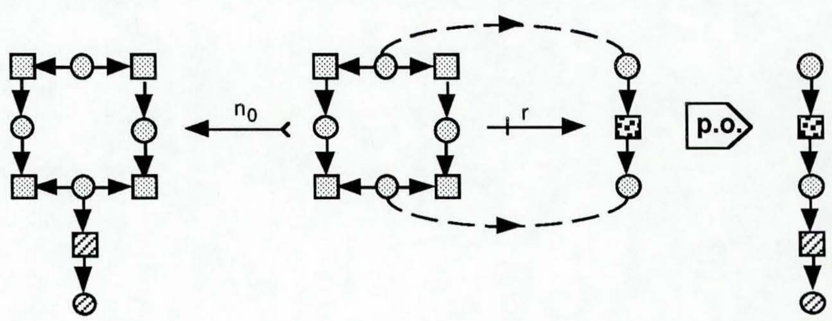
a) expansion: transforms part of a net (usually a transition) into a possible more complex net. An expansion represents a change from a higher level into a lower level of abstraction, such as in the top-down design of systems;
b) abstraction: it is the opposite of expansion, as in a bottom-up design of systems;
c) addition: adds states and transition to a net, possibly identifying some parts (which already exist in the net);
d) deletion: deletes parts of a net;
e) mix: neither of the above cases.

Example 3.3 In each item below, consider the diagram constituted by the rule r and the instantiation n_0. The resulting net is determined by the pushout construction of the diagram where dashed boxes and circles identify those parts which are preserved by morphisms:

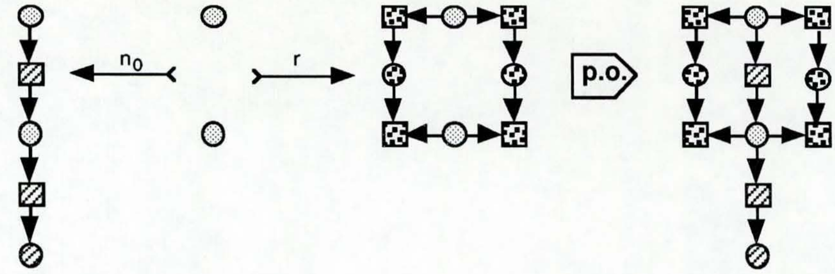
a) Expansion: a single arc is further detailed into four arcs:



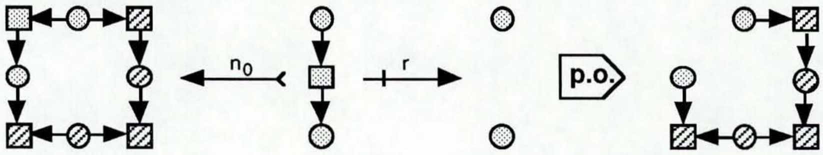
b) Abstraction: a sub-net with four arcs (and four nodes) is abstracted into only one arc (and two nodes):



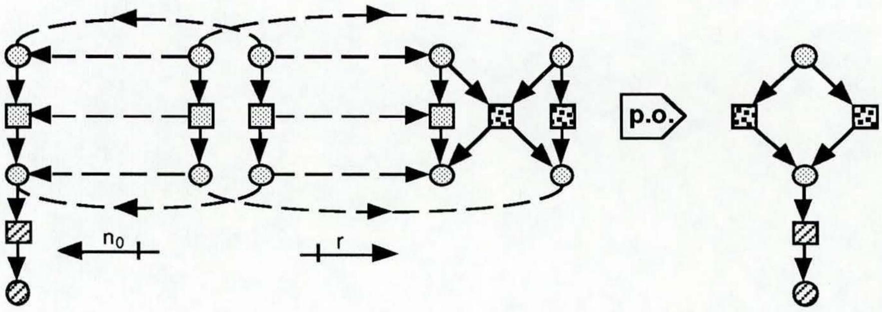
c) Addition: adds four transitions and two states (and identify two states):



d) Deletion: deletes part of a net, preserving the shape of the remaining part:



e) Mix: in the above examples, the instantiation morphisms are total and injective. However, it can be a partial morphism of any kind:



❑

3.2 Reification of Petri Nets

Consider a rule r, an instantiation n_0 and the resulting transformation φ. Since, φ is a net morphism, by definition, it is also a rule. Also, it is straightforward to prove that the diagram below is a pushout. Thus, any *pMPetri*-morphism is both a rule and a transformation.

pMPetri

$$
\begin{tikzcd}
 & N \arrow[dl, "id_N"'] \arrow[dr, "\varphi"] & \\
N \arrow[dr, "\varphi"'] \arrow[rr, phantom, "\text{p.o.}"] & & M \arrow[dl, "id_M"] \\
 & M &
\end{tikzcd}
$$

Consider the rules r: $N_0 \to M_0$, s: $P_0 \to Q_0$, the instantiations n_0: $N_0 \to N$, p_0: $P_0 \to M$ and the transformations φ, ψ illustrated in the diagram below. The composition of $\psi \circ \varphi$ should also be given by a pushout with rule r' and instantiation n_0' determined by r, s, n_0, p_0. In fact, there are many rules and instantiations which satisfy this requirement. But, since φ and ψ are also rules (determined by r, s, n_0, p_0), a very simple pushout which results in the composed transformation $\psi \circ \varphi$ is given by the rule $r' = \psi \circ \varphi$ and the instantiation $n_0' = id_N$.

$$
\begin{tikzcd}
N_0' \arrow[rrrrrr, "r'"] \arrow[ddr, bend right, "n_0'"'] & & & & & & Q_0' \arrow[ddl, dashed] \\
& N_0 \arrow[r, "r"] \arrow[d, "n_0"] \arrow[drr, phantom, "\text{p.o.}"] & M_0 \arrow[dr] & & P_0 \arrow[r, "s"] \arrow[dl, "p_0"'] \arrow[d, phantom, "\text{p.o.}"] & Q_0 \arrow[d] & \\
& N \arrow[rr, "\varphi"] \arrow[rrrr, bend right, "\phi = \psi \circ \varphi"'] & & M \arrow[rr, "\psi"] & & Q &
\end{tikzcd}
$$

pMPetri

Therefore, a transformation morphism φ: $N \to M$ is fully determined by a pair $\langle r, n_0\rangle$ where r: $N_0 \to M_0$ is a rule and n_0: $N_0 \to N$ is an instantiation. However, φ may also be determined by other pairs such as $\langle \varphi, id_N\rangle$. Thus, we may consider classes of equivalence of pairs of morphisms with respect to the relation "the transformations determined by the pushouts coincide". A class of equivalence is called a reification. Petri nets as objects and reifications as morphisms constitute the category *rMPetri*.

Definition 3.4 Category of Petri Nets and Reifications. Consider the category *pMPetri*. The category *rMPetri* is defined as follows:

a) *rMPetri* has the same objects as *pMPetri*.

b) A morphisms in *rMPetri*, called reification, is an equivalence class of pairs of morphisms $\langle r: N_0 \to M_0, n_0: N_0 \to N\rangle: N \to M$ with respect to the relation $\langle r: N_0 \to M_0, n_0: N_0 \to N\rangle$ reif $\langle r': N_0' \to M_0', n_0': N_0' \to N\rangle$ if and only if the resulting pushouts determine the following commutative diagram:

$$
\begin{tikzcd}
N_0 \arrow[r, "n_0"] \arrow[d, "r"'] \arrow[dr, phantom, "\text{p.o.}"] & N \arrow[d, dashed, "\varphi"] \arrow[dr, phantom, "\text{p.o.}"] & N_0' \arrow[l, "n_0'"'] \arrow[d, "r'"] \\
M_0 \arrow[r] & M & M_0' \arrow[l]
\end{tikzcd}
$$

pMPetri

A class $[\langle r, n_0\rangle]$: $N \to M$ may be denoted by a representative element $\langle r, n_0\rangle$ or by the transformation morphism φ: $N \to M$ which defines the class. The identity reification ι_N: $N \to N$ is the equivalence class $[\langle id_N, id_N\rangle]$: $N \to N$.

c) The composition of φ: $N \to M$, ψ: $M \to Q$, denoted by $\psi\circ\varphi$: $N \to Q$, is the class $[\langle \psi\circ\varphi, id_N\rangle]$: $N \to Q$. ❑

3.3 Vertical Composition

In the next proposition, we prove that the categories *rMPetri* and *pMPetri* are isomorphic. Thus, the vertical compositionality of Petri nets with respect to the reification is a direct corollary. In what follows, note that, for any class $[\langle r, n_0\rangle]$: $N \to M$ where $\langle r, n_0\rangle$ is a representative element, the pair $\langle \varphi_{r,n_0}, id_N\rangle$ is also an element of the class.

Proposition 3.5 The categories *rMPetri* and *pMPetri* are isomorphic.

Proof: Consider:

a) the functor *pr: pMPetri* $\to$ *rMPetri* such that for all net P and for all φ: $N \to M$ and ψ: $M \to Q$:

$$pr P = P,\ pr\, id_P = [\langle id_P, id_P\rangle],\ pr\varphi = [\langle \varphi, id_N\rangle] \text{ and } pr(\psi\circ\varphi) = [\langle \psi, id_M\rangle]\circ[\langle \varphi, id_N\rangle] = [\langle \psi\circ\varphi, id_N\rangle].$$

b) the functor *rp: rMPetri* $\to$ *pMPetri* such that for all net P and for all φ: $N \to M$ and ψ: $M \to Q$:

$$rp P = P,\ rp[\langle id_P, id_P\rangle] = id_P,\ rp[\langle \varphi, id_N\rangle] = \varphi \text{ and } rp[\langle \psi\circ\varphi, id_N\rangle] = \psi\circ\varphi.$$

Then $rp \circ pr = id_{pMPetri}$ and $pr \circ rp = id_{rMPetri}$. ❑

Since *rMPetri* and *pMPetri* are isomorphic the composition of reifications is straightforward and thus, the *vertical compositionality is achieved.* Also, we identify both categories by *pMPetri* and use the terms reification and transformation indifferently. A morphism φ: $A \to B$ which is a reification may also be represent as φ: $A \Rightarrow B$.

3.4 Horizontal Compositionality

In the following proposition, we prove that the horizontal compositionality of Petri nets is achieved, i.e., the reification of nets distributes through the parallel composition (categorical product) of component nets.

Proposition 3.6 Let $\{\varphi_i: N_i \Rightarrow M_i\}_{i\in I}$ be an arbitrary indexed set of *pMPetri*-reifications, where I is a set. Then $\times_{i\in I}\varphi_i: \times_{i\in I} N_i \Rightarrow \times_{i\in I} M_i$.

Proof: Since *pMPetri* is complete, $\times_{i\in I}\varphi_i: \times_{i\in I} N_i \Rightarrow \times_{i\in I} M_i$ is the morphism uniquely induced by the product construction in *pMPetri*, as follows:

$$\begin{tikzcd} N_i \arrow[d, "\varphi_i"'] & \times_{i\in I} N_i \arrow[l, "\pi_{N_i}"'] \arrow[d, dotted, "\times_{i\in I}\varphi_i"] \arrow[dl, dashed, bend right] \\ M_i & \times_{i\in I} M_i \arrow[l, "\pi_{M_i}"] \end{tikzcd}$$

❑

4 Specification of Petri Nets

Usually, for some given system, we want to specify only a set of possible reifications in order to obtain the desired derived system. Since, until now, the category defined has all possible reifications, we introduce two techniques which are

- specification grammar and the corresponding subcategory of reifications, reflecting all desired derivations of a given system;
- hierarchical specification which is a generalization of the top down approach, where the reifications are restricted in such way that it is not possible to substitute a part of a net which is only partially substituted by the previous change of level of abstraction.

4.1 Specification Grammar

A specification grammar is basically an initial net and a collection of possible rules and instantiations. Each grammar induces a subcategory of *pMPetri* which reflects the possible derivations from the initial net. Thus, a grammar can be considered as the specification a system and the induced subcategory as the levels of abstractions of the system and their relationship.

Definition 4.1 Specification Grammar. A specification grammar or just grammar is a triple $\mathrm{Gram} = \langle R, I, N\rangle$ where R, I are collections of *pMPetri*-morphisms representing the rules and instantiations of the grammar and N is an *pMPetri*-object called initial net. ❑

Each grammar induces a subcategory of *pMPetri* with all nets that can be derived from the initial one using the given rules and instantiations.

Definition 4.2 Subcategory Induced by a Grammar. Let $\mathrm{Gram} = \langle R, I, N\rangle$ be a grammar. The subcategory *Gram* of *pMPetri* induced by the grammar Gram is inductively defined as follows:

a) N is an *Gram*-object and $[\langle id_N, id_N\rangle]: N \to N$ is a *Gram*-morphism;
b) for all *Gram*-object M, for all instantiation $m_0: M_0 \to M$ and for all rule $r: M_0 \to P_0$, $[\langle r, m_0\rangle]: M \to P$ is a *Gram*-morphism and P is an *Gram*-object;
c) for all *Gram*-morphisms $\varphi: M \to P$, $\psi: P \to Q$, the morphism $[\langle \psi\circ\varphi, id_M\rangle]: M \to Q$ is a *Gram*-morphism. ❑

4.2 Top Down Design

The top down design of nets can be achieved restricting the composition of reifications to two possible cases: independent and total dependent. Two reifications that can be composed and are not related by one of this cases are not allowed.

Definition 4.3 Independent and Total Dependent Reifications. Consider the reifications $\varphi_{s,n_0}N = M$, $\psi_{r,p_0}M = Q$ represented in the figure below.

a) The reifications φ and ψ are independent if there is an instantiation $p_i: P_0 \to N$ such that $p_0 = \varphi \circ p_i$. In this case,

ψ transforms parts of M exclusively inherited from N.

b) The reifications φ and ψ are total dependent if there is an instantiation p_d: $P_0 \rightarrow M_0$ such that $p_0 = p_d \circ m_0$. In this case, ψ substitutes parts of M exclusively inherited from M_0.

$$\begin{tikzcd}[column sep=large]
N_0 \arrow[r, "s"] \arrow[d, "n_0"'] \arrow[dr, phantom, "\text{p.o.}"] & M_0 \arrow[d, "m_0"] & P_0 \arrow[l, dashed, "p_d"'] \arrow[r, "r"] \arrow[dl, "p_0"] \arrow[dll, dashed, "p_i"'] \arrow[dr, phantom, "\text{p.o.}"] & Q_0 \arrow[d, "q_0"] \\
N \arrow[r, "\varphi"] & M \arrow[rr, "\psi"] & & Q
\end{tikzcd}$$

pMPetri

❑

For the independent reifications, the composition may be defined as the simultaneous derivation of the component reifications. In the case of total depend reifications, the composition is, basically the sequential composition of the rules.

Proposition 4.4 Consider the rules s: $N_0 \rightarrow M_0$, r: $P_0 \rightarrow Q_0$ and the instantiations n_0: $N_0 \rightarrow N$, p_0: $P_0 \rightarrow M$ such that $\varphi_{(s,n_0)}N = M$ and $\psi_{(r,p_0)}M = Q$.

a) For the independent reifications, the composition of pushouts is defined as follows:

$$\begin{tikzcd}
& N_0+P_0 \arrow[dl, "n_0+p_i"'] \arrow[dr, "s+r"] \arrow[dd, phantom, "\text{p.o.}"] & \\
N \arrow[dr, "\psi\circ\varphi"'] & & M_0+Q_0 \arrow[dl] \\
& Q &
\end{tikzcd}$$

pMPetri

where $s+r$ and n_0+p_i are uniquely induced by the coproduct construction in *pMPetri* as follows:

$$\begin{tikzcd}
P_0 \arrow[rrrrr, "r"] \arrow[dr] \arrow[dddrr, bend right, "p_i"'] & & & & & Q_0 \arrow[dl] \arrow[ddd, "q_0"] \\
& N_0+P_0 \arrow[rrr, dashed, "s+r"] \arrow[ddr, dashed, "n_0+p_i"'] & & & M_0+Q_0 \arrow[ddr, dashed, "m_0+q_0"] & \\
& & N_0 \arrow[ul] \arrow[r, "s"] \arrow[d, "n_0"] & M_0 \arrow[ur] \arrow[d, "m_0"] & & \\
& & N \arrow[r, "\varphi"] & M \arrow[rr, "\psi"] & & Q
\end{tikzcd}$$

b) For the total dependent derivations, the composition of pushouts is defined as follows:

$$\begin{tikzcd}
& N_0 \arrow[dl, "n_0"'] \arrow[dr, "r'\circ s"] \arrow[dd, phantom, "\text{p.o.}"] & \\
N \arrow[dr, "\psi\circ\varphi"'] & & Q'_0 \arrow[dl, "q'_0"] \\
& Q &
\end{tikzcd}$$

pMPetri

where the morphisms r' and q_0' are induced as follows (③ is a pushout):

$$\begin{tikzcd}
& P_0 \arrow[rr, "r"] \arrow[d, "p_d"'] \arrow[drr, phantom, "\textcircled{3}"] & & Q_0 \arrow[dl, "p_d'"'] \arrow[dd, "q_0"] \\
N_0 \arrow[r, "s"] \arrow[d, "n_0"'] \arrow[dr, phantom, "\textcircled{1}"] & M_0 \arrow[r, "r'"] \arrow[d, "m_0"] \arrow[dr, phantom, "\textcircled{4}"] & Q_0' \arrow[dr, dashed, "q_0'"'] & \\
N \arrow[r, "\varphi"] & M \arrow[rr, "\psi"] & & Q \quad \textcircled{2}
\end{tikzcd}$$

Proof: The case (a) is the sequential independent case proposed in [Löwe 90]. For the case (b) we have only to prove that the external diagram determined by ① and ④ is a pushout. Let $u: M \to W$ and $v: Q_0' \to W$ be morphisms such that $u \circ m_0 = v \circ q_0'$. Since ② is a pushout, there is a unique $w: Q \to W$ such that $u = w \circ \psi$ and $v \circ p_d' = w \circ q_0$. Since ③ is a pushout, there is a unique $v': Q_0' \to W$ such that $u \circ m_0 = v' \circ r'$ and $v' \circ p_d' = w \circ q_0$. Then, $v' = v$. Since $v \circ p_d' = w \circ q_0$ and $q_0 = q_0' \circ p_d'$ then $v \circ p_d' = w \circ q_0' \circ p_d'$. By the uniqueness of v, $v = w \circ q_0'$. Suppose that there is $w': Q \to W$ such that $u = w' \circ \psi$ and $v = w' \circ q_0$. By the uniqueness of w, $w' = w$. Since ① and ④ are pushouts, then the external diagram of ① and ④ is also a pushout. ❑

4.3 Black Box Reification

The black box reification is the reification of single transitions satisfying some constrains. It is determined by a grammar such that:

- each rule replaces a single transitions;
- for each transition reified, the remaining part of the net (including the source and target places of the replaced transition) stay unchanged;
- for each transition reified, the new part of the net interact with the remaining one only through the inherited places of the replaced transition. Also, the new part simulates the source and target functions of the replaced transition in an "atomic way". For instance, if the firing of the replaced transition consumes n tokens of some place, the new part consumes n tokens at once in the corresponding inherited place.

If a reification replaces a transition by several new transitions we can not expect that the original net and the resulting one are equivalent, according to some notion of direct simulation between component transitions. Following the same idea, if a reification introduces new places we can not expect an equivalence between the reified and unreified nets according to some notion of observation of state changes. However, if we consider the new part as a "black box" where, except by the inherited places, the included transitions and places are considered as hidden, then it is possible the define some kind of direct simulation between the reified transition and the black box. This notion of simulation can be extended to the unreified and reified nets.

First we introduce the concepts used to define a black box reification and the corresponding bisimulation. In a Petri net, the change from a marking m into a marking m' when the transition t fires, is denoted as follows:

$$m \xrightarrow{t} m'$$

Definition 4.5 Places of a Transition. Consider a *pMPetri*-object $M = \langle V^{\oplus}, T, \partial_0, \partial_1, I, init\rangle$. Let t be a transition in T such that $\partial_0(t) = u_1A_1 \oplus u_2A_2 \oplus \ldots \oplus u_kA_k$ and $\partial_1(t) = v_1B_1 \oplus v_2B_2 \oplus \ldots \oplus v_nB_n$. Then:

a) $places(\partial_0(t)) = \{A_1, A_2, \ldots, A_k\}$ and $places(\partial_1(t)) = \{B_1, B_2, \ldots, B_n\}$;

a) $places(t) = \{A_1, A_2, \ldots, A_k, B_1, B_2, \ldots, B_n\}$. ❑

Definition 4.6 Path. Consider a *pMPetri*-object M. A path of M from the marking m_0 is a (possibly empty) sequence of transitions $t_1, t_2, t_3, \ldots$ such that:

$$m_0 \xrightarrow{t_1} m_1 \xrightarrow{t_2} m_2 \xrightarrow{t_3} \cdots$$

where the last transition of the sequence (if exists) leads to a marking from which there is no transition able to fire. In this case, the path is called finite. The set of all possible paths of M from some given marking m_0 is denoted by $path(M, m_0)$. ❑

Definition 4.7 Reachable Markings. Consider a *pMPetri*-object M. Then:

a) a marking m_n is reachable from a marking m_0, if there is a (possibly empty) sequence of transitions $t_1, t_2, \ldots, t_n$ such that:

$$m_0 \xrightarrow{t_1} m_1 \xrightarrow{t_2} \cdots \xrightarrow{t_n} m_n$$

The set of all reachable markings from a marking m_0 is denoted by $\mathrm{reach}(M, m_0)$;

b) a marking m_n is reachable by a path $p = t_1, t_2, \ldots, t_n$ from m_0 if there is a sequence of markings $m_1, m_2, \ldots, m_{n-1}$ such that:

$$m_0 \xrightarrow{t_1} m_1 \xrightarrow{t_2} \cdots \xrightarrow{t_{n-1}} m_{n-1} \xrightarrow{t_n} m_n$$

The set of all the markings reachable by p from m_0 is denoted by $\mathrm{reachpath}(p, m_0)$. ❑

Example 4.8 Consider the *pMPetri*-object $M = \langle\{A, B, C, D\}^{\oplus}, \{u, v\}, \partial_0, \partial_1, I, \mathrm{init}\rangle$ with $u: 2A \to B \oplus C$ and $v: C \to D$. Then:

a) $\mathrm{places}(\partial_0(u)) = \{A\}$, $\mathrm{places}(\partial_1(u)) = \{B, C\}$ and $\mathrm{places}(u) = \{A, B, C\}$;

c) $\mathrm{path}(M, 5A) = \{\langle u, u, v, v\rangle, \langle u, v, u, v\rangle\}$;

c) $\mathrm{reach}(M, 5A) = \{5A, 3A \oplus B \oplus C, A \oplus 2B \oplus 2C, 3A \oplus B \oplus D, A \oplus 2B \oplus 2D\}$ and $\mathrm{reachpath}(\langle u, u, v, v\rangle, 5A) = \{A \oplus 2B \oplus 2D\}$. ❑

Black Box Reification

Definition 4.9 Black Box Rule, Instantiation, Reification, Grammar. Consider the nets $N_t = \langle V_t^{\oplus}, \{t\}, \partial_{t0}, \partial_{t1}, I_t, \mathrm{init}_t\rangle$ and $M_0 = \langle V_0^{\oplus}, T_0, \partial_{00}, \partial_{01}, I_0, \mathrm{init}_0\rangle$.

a) A rule $r = \langle r_V, r_T, r_I\rangle: N_t \to M_0$ is a black box rule if:

a.1) r_T is not defined for the transition t;

a.2) r_V is induced by a total and injective function in the carriers of the semi-groups;

a.3) r_I is a total and injective function;

a.4) there are unique transitions ✪ and ✞ in T_0, named fork and joint, respectively, such that:

$$\mathrm{places}(\partial_{t0}(t)) = \mathrm{places}(\partial_{00}(✪)) \text{ and } \mathrm{places}(\partial_{t1}(t)) = \mathrm{places}(\partial_{01}(✞)).$$

Also, fork and joint must satisfy:

$$r_V(\partial_{t0}(t)) = \partial_{00}(✪) \text{ and } r_V(\partial_{t1}(t)) = \partial_{01}(✞).$$

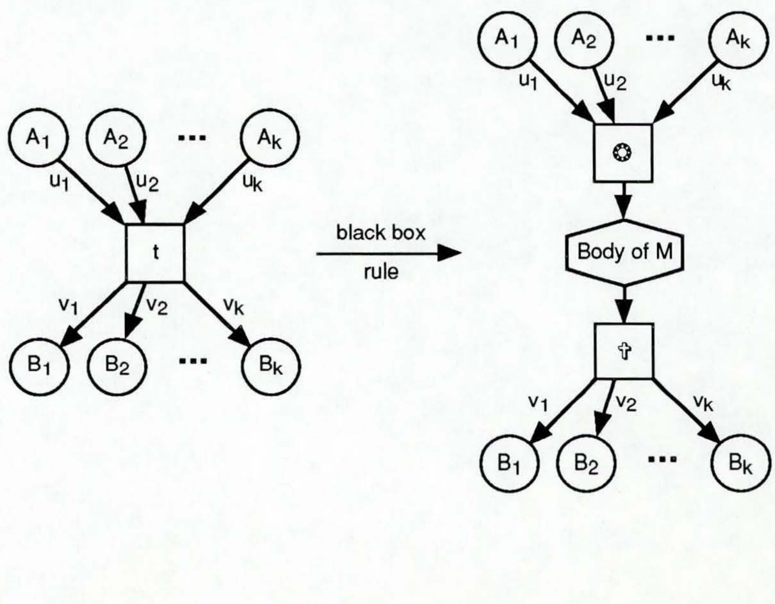
b) An instantiation $n_t = \langle n_V, n_T, n_I\rangle: N_t \to N$ is a black box instantiation if n_V, n_T, n_I are total and monic.

c) A reification φ_{r,n_t} is a black box reification if r is a black box rule and n_t is a black box instantiation.

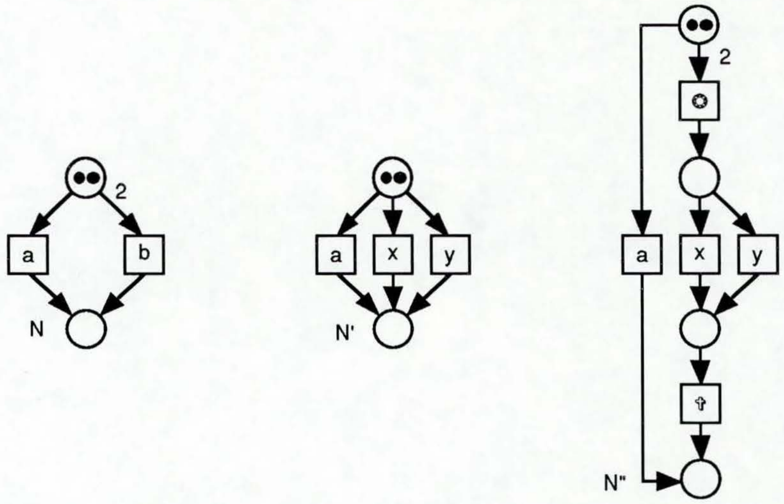
d) A grammar $\mathrm{Gram} = \langle R, I, N\rangle$ is a black box (specification) grammar if R is a collection of black box rules and I is a collection of black box instantiations. ❑

Note that the source nets of a black box rule and instantiations have only one transition. Also, for a black box rule we have that (see figure below):

- item a.1) state that the single transition of the source net is replaced;
- item a.2) specify that the "shape" of places of the source net are preserved;
- item a.4) ensures that tokens consumed (produced) by t are consumed (produced) by M at once. If desired, the uniqueness of the fork and joint transitions can be relaxed.

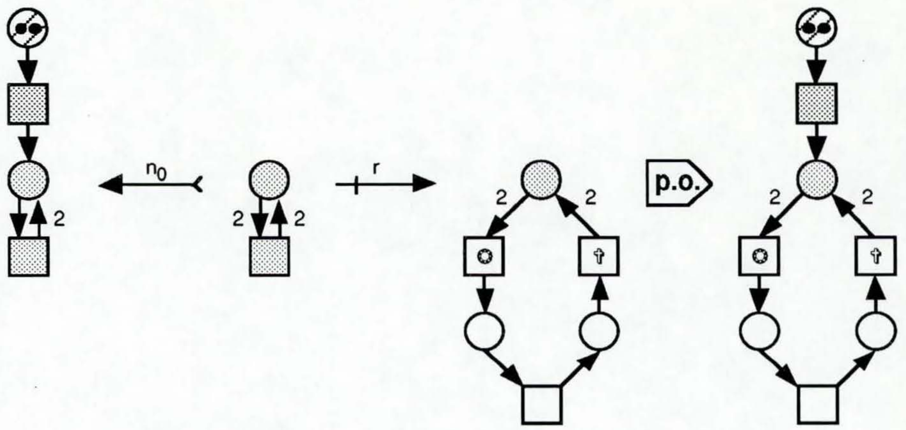


To illustrate the idea of the fork transition, consider the figure below. In the net N, if b fires, then a is not able to fire. Suppose that N is reified into N', where the transition b is replaced by the transitions x, y. In this case, if x fires, a is able to fire. Note that, if N is reified into N", we guaranteed that when fork fires a is not able to fire. The idea of the joint transition is analogous.

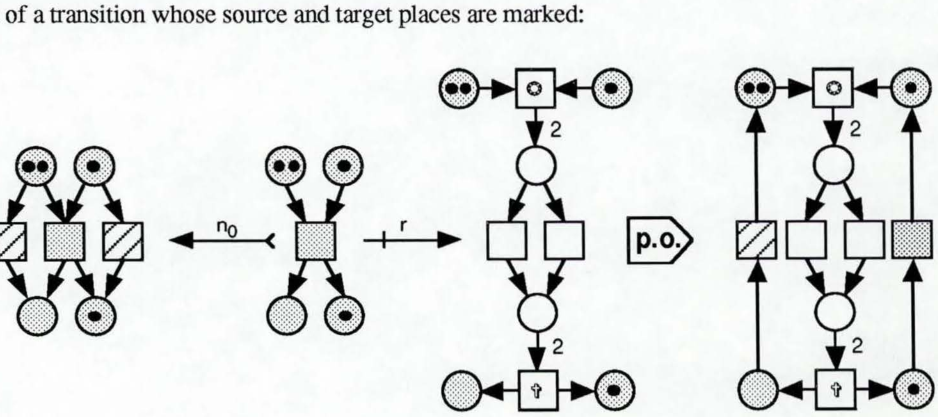


Example 4.10 In each item below, consider the diagram constituted by the black box rule r and the black box instantiation n_0. The resulting net is determined by the pushout construction of the diagram where dashed boxes and circles identify the parts preserved by morphisms:

a) reification of an idle transition:



b) reification of a transition whose source and target places are marked:



❑

Simulation and Bisimulation

Consider a black box rule $r = \langle r_V, r_T, r_I \rangle: N_t \to M$. Since r_V, r_I are monic and r_T is undefined for the unique transition in N_t, $r^{-1} = \langle r_V^{-1}, r_T^{-1}, r_I^{-1} \rangle: M \to N_t$ is also a *pMPetri*-morphism.

Definition 4.11 r-Simulation and r-Bisimulation. Consider the nets $N_t = \langle V_t^{\oplus}, \{t\}, \partial_{t0}, \partial_{t1}, l_t, init_t \rangle$ and $M_0 = \langle V_0^{\oplus}, T_0, \partial_{00}, \partial_{01}, l_0, init_0 \rangle$ and a black box rule $r = \langle r_V, r_T, r_I \rangle: N_t \to M_0$. Then:

a) M_0 r-simulates N_t, denoted by $N_t \xrightarrow{r} M_0$, if and only if:
   - a.1) for all m in $V_t^{\oplus}$ and for all p in $path(M_0, r_V(m))$, p is finite;
   - a.2) for all m in $V_t^{\oplus}$ and for all p in $path(M_0, r_V(m))$, $reach(N_t, m) = reachpath(p, r_V(m))$;

b) N_t r-simulates M_0, denoted by $N_t \xleftarrow{r} M_0$, if and only if:
   - b.1) for all m in $V_0^{\oplus}$ and for all p in $path(M_0, m)$, p is finite;
   - b.2) for all m in $V_0^{\oplus}$ and for all p in $path(M_0, m)$, $reachpath(p, m) = reach(N_t, r_V^{-1}(m))$;

c) N_t r-bisimulates M_0, denoted by $N_t \xleftrightarrow{r} M_0$, if and only if M_0 r-simulates N_t and N_t r-simulates M_0. ❑

In the above definition, a.1), b.1) ensure that the replaced transition is simulated in a finite time. Items a.2), b.2) ensure that, for the "same" source marking, N_t and M_0 stop with the "same" target marking. Also, from the definition of black box rule, a.2), b.2) imply that M_0 always stops after firing the joint transition.

Definition 4.12 Simulation and Bisimulation. Consider the nets N_t and M_0, a black box rule $r: N_t \to M_0$, a black box instantiation $n_t: N_t \to N$ and the resulting reification $\varphi_{r,n_t}: N \to M$. Then:

a) M φ-simulates N, denoted by $N \xrightarrow{\varphi} M$, if and only if $N_t \xrightarrow{r} M_0$;
b) N φ-simulates M, denoted by $N \xleftarrow{\varphi} M$, if and only if $N_t \xleftarrow{r} M_0$;
c) N φ-bisimulates M, denoted by $N \xleftrightarrow{\varphi} M$, if and only if $N_t \xleftrightarrow{r} M_0$; ❑

5 Concluding Remarks

The graph transformation concept based on the single pushout approach is extended for Petri nets with partial morphisms where classes of transformations stand for reifications of nets. In this context, the composition of reifications (i.e., the composition of pushouts) is defined leading to categories of nets and reifications. Thus, reification of nets compose and the vertical compositionality is achieved without modifying or adding extra structure to the basic definition of Petri nets. Then we show that the reification distributes over the parallel composition (categorical product) of nets and the horizontal compositionality is also achieved meaning that, in the proposed framework, we are able to further specify a concurrent system before of after the joint behaviour of its component parts in order to obtain the same resulting system.

Also, we introduce two technique for system specification namely specification grammar which allows the definition of all desired reifications for some given net and hierarchical specification which generalizes the top-down design of systems.

Finally, we provide the black box reification, where transitions are replaced by nets, satisfying some constrains. In this case, we are able to define a kind of bisimulation between the unreified and the reified nets, provided that the part introduced is viewed as a "black box", where everything is hidden except the inherited source and target places of the transition replaced.

Now, we are working on several notions of a "correct" reification, the corresponding notion of simulation and bisimulation between unreified and reified nets and the preservation of properties such as liveness and safeness.

Acknowledgments

The author is grateful to Amílcar Sernadas, Carlos Caleiro and Pedro Resende for many fruitful discussions.

References

[Adámek *et al* 90] J. Adámek, H. Herrlich, and G. Strecker, *Abstract and Concrete Categories*, Wiley, 1990.

[Asperti and Longo 91] A. Asperti and G. Longo, *Categories, Types and Structures - An Introduction to the Working Computer Science* in Foundations of Computing, M. Garey and A. Meyer (eds.), MIT Press, 1991.

[Barr and Wells 90] M. Barr and C. Wells, *Category Theory for Computing Science*, Prentice Hall, 1990.

[Broek 91] P.M. van den Broek, *Algebraic Graph Rewriting Using a Single Pushout*, TAPSOFT '91, LNCS 493, Springer-Verlag, 1991

[Brown *et al* 91] C. Brown, D. Gurr and V. de Paiva, *A Linear Specification Language for Petri Nets*, Technical Report DAIMI PB-363, Arhus University, 1991.

[Casley 91] R. T. Casley, *On the Specification of Concurrent Systems*, Ph.D. thesis, University of Stanford, 1991.

[Corradini 90] A. Corradini, *An Algebraic Semantics for Transition Systems and Logic Programming*, Ph.D. thesis, technical report TD-8/90, Università di Pisa, 1990.

[Ehrig 79] H. Ehrig, *Introduction to the Algebraic Theory of Graph Grammars*, LNCS 93, 1979.

[Gorrieri 90] R. Gorrieri, *Refinement, Atomicity and Transactions for Process Description Language*, Ph.D. Thesis, Università di Pisa, 1990.

[Löwe 90] M. Löwe, *Algebraic Approach to Graph Transformation Based on a Single Pushout Derivation with Partial Morphisms*, Techn. Report 90/5, Technische Universitat Berlin, 1990.

[Löwe and Ehrig 90] M. Löwe and H. Ehrig, *Algebraic Approach to Graph Transformation Based on a Single Pushout Derivation*, IEEE Trans. Software Engineering SE 3, 1990.

[Mac Lane 71] S. Mac Lane, *Categories for the Working Mathematician*, Springer-Verlag, 1971.

[Menezes and Costa 93] P. B. Menezes and J. F. Félix, *Synchronization in Petri Nets*, technical report INESC/RT/71-93, preprint IST/2-94, INESC/IST, Lisbon, 1993, submitted.

[Menezes 93] P. B. Menezes, *Algebraic Implementation of Transition Systems*, technical report INESC/RT/72-93, preprint IST/3-94, INESC/IST, Lisbon, 1993, submitted.

[Menezes 94] P. B. Menezes, *Reification of Petri Nets through Graph Transformations*, INESC/IST, Lisbon, 1993, submitted.

[Meseguer and Montanari 90] J. Meseguer and U. Montanari, *Petri Nets are Monoids*, Information and Computation 88, pp. 105-155, Academic Press, 1990.

[Milner 89] R. Milner, *Communication and Concurrency*, Prentice Hall, 1989.

[Reisig 85] W. Reisig, *Petri Nets: An Introduction*, EATCS Monographs on Theoretical Computer Science 4, Springer-Verlag, 1985.

[Reisig 92] W. Reisig, *A Primer in Petri Net Design*, , Springer-Verlag, 1992.

[Vogler 92] W. Vogler, *Modular Construction and Partial Order Semantics of Petri Nets*, LNCS 625, Springer-Verlag, 1992.

[Winskel 87] G. Winskel, *Petri Nets, Algebras, Morphisms and Compositionality*, Information and Computation 72, pp. 197-238, Academic Press, 1987.

[Winskel and Nielsen 94] G. Winskel and M. Nielsen, *Models for Concurrency*, draft copy in 1994, to appear.