A representation of processes of Petri nets by matrices *

Józef Winkowski
Instytut Podstaw Informatyki PAN
Ordona 21, 01-237 Warszawa, Poland
wink@ipipan.waw.pl

1997.5.27

Abstract. A correspondence between processes of Petri nets and partitioned matrices over freely generated semirings is described. This correspondence implies a correspondence between operations of composing processes sequentially and in parallel and operations of multiplying and juxtaposing the respective partitioned matrices. It results in a characterization of partitioned matrices corresponding to processes of a given net and, consequently, allows one to represent processes by their matrices.

Keywords: Petri net, process, sequential composition, parallel composition, interchange, partitioned matrix, multiplication, juxtaposition.

1. Problem

Place/Transition Petri nets, or briefly nets, are bipartite graphs representing concurrent systems (cf. [GLT 80], [R 85], [GV 87], [Wns 87], [MM 88], and the definition of nets in section 2, for details). A graph of this type is shown in figure 1. Nodes depicted as circles represent places in which some resources called tokens may reside. Those depicted as boxes represent transitions which when executed consume tokens from places and produce tokens in places as indicated by directed edges.

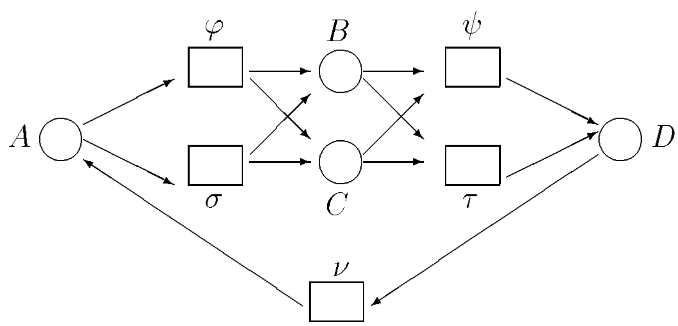

Figure 1

*This work has been partially supported by grant 8T11C 029 08. It has been published in Fundamenta Informaticae 30 (1) (1997) 97-107.

Processes of a net are isomorphism classes of unfoldings of this net into acyclic nets without branching at places, each unfolding representing a concrete execution of the respective concurrent system or a segment of such an execution (cf. [GR 83], [BD 87], [DMM 89], and the definition of net processes in section 2, for details). An unfolding of the net in figure 1 and the corresponding process are shown in figures 2 and 3, respectively. Circles represent tokens taking part in the considered execution, each token in a place indicated by the respective label. Boxes represent concrete executions of transitions indicated by the respective labels, each execution consuming and producing concrete tokens as indicated by directed edges.

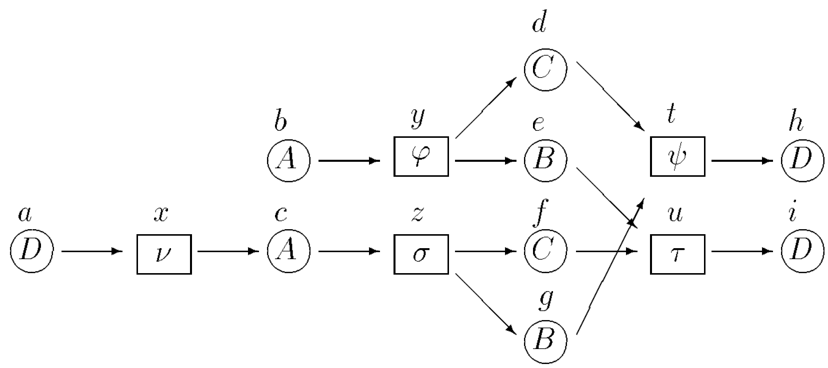

Figure 2

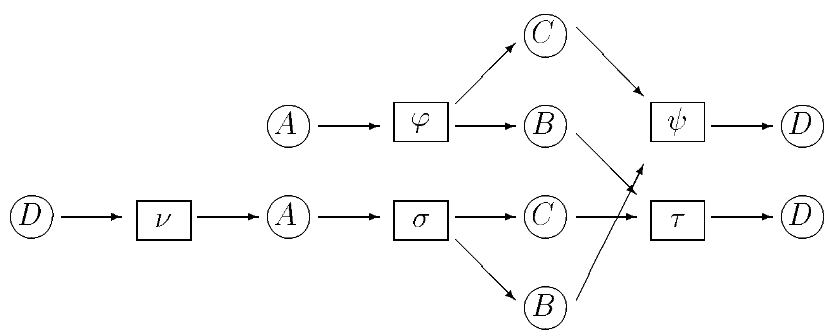

Figure 3

To each finite process of a net a partitioned matrix similar to causal streams of [FMM 91] corresponds whose items describe how resulting tokens of this process have been obtained from initial ones by executing sequences of transitions (cf. [FMM 91] and [Wnk 95]). The matrix corresponding to the process in figure 3 is shown in figure 4. This matrix is partitioned by names of places into two submatrices corresponding to its rows.

	D	D
A	$\varphi\psi$	$\varphi\tau$
D	$\nu\sigma\psi$	$\nu\sigma\tau$

Figure 4

Partitioned matrices corresponding to processes of a net can be regarded as members of a universe of partitioned matrices. It is interesting how to distinguish them from members which do not correspond to processes, and what information they contain on processes they correspond to.

In the present paper, that is an extension of [Wnk 96], we give a solution to this problem. The role of this solution lies in the fact that it allows one to represent processes of a net as partitioned matrices with special properties rather than as unfoldings, which is convenient for algebraic treatment.

2. Formalization

A *Place/Transition Petri net* (or briefly a *net*) can be defined as $N = (P, T, pre, post)$, where P is a set of *places*, T such that $P \cap T = \emptyset$ is a set of *transitions*, and $pre, post : T \to P^{\oplus}$, where $P^{\oplus}$ is the free commutative monoid generated by the set P of places, are respectively a *consumption function* and a *production function* (cf. [MM 88]). Each $\pi \in P^{\oplus}$ can be regarded as a multiset of places with a multiplicity $\pi(p)$ of each place p. For each transition t, the multiset $pre(t)$ represents a collection of tokens, $(pre(t))(p)$ tokens in each place p, which must be consumed in order to execute t, and the multiset $post(t)$ represents a collection of tokens, $(post(t))(p)$ tokens in each place p, which is produced by executing t. For the purpose of this paper it is necessary to assume that $pre(t) \neq 0$ and $post(t) \neq 0$ for all $t \in T$.

We denote by E the set $P \cup T$ and we use subscripts, E_N, P_N, T_N, pre_N, $post_N$, when necessary to avoid a confusion.

A *morphism* from a net N to a net N' can be defined as a triple $m : N \to N'$, where m is a mapping from E_N to $E_{N'}$ such that $m(P_N) \subseteq P_{N'}$, $m(T_N) \subseteq T_{N'}$, and the unique extension of m to a monoid homomorphism $m^{\oplus} : P_N^{\oplus} \to P_{N'}^{\oplus}$ satisfies the conditions $m^{\oplus}(pre_N(t)) = pre_{N'}(m(t))$ for all $t \in T_N$ and $m^{\oplus}(post_N(t)) = post_{N'}(m(t))$ for all $t \in T_N$.

According to [DMM 89] the behaviour of a net N can be represented by the set of its *concatenable processes* (called in the sequel *processes*), a process being an equivalence class of *process instances* of the form $A = (X, Y, pred, suc, m, i, j)$ such that

(1) $N_A = (X, Y, pred, suc)$ is a finite net with the following properties (a *process net*):

  (1.1) $(pred(y))(x) \leq 1$ and $(suc(y))(x) \leq 1$ for each $x \in X$ and $y \in Y$ (there is at most one edge from $x \in X$ to $y \in Y$ and at most one edge from $y \in Y$ to $x \in X$),

  (1.2) $(pred(y))(x) = (pred(y'))(x) = 1$ implies $y = y'$ and $(suc(y))(x) = (suc(y'))(x) = 1$ implies $y = y'$ for each $x \in X$ and $y, y' \in Y$ (there is no branching at places),

  (1.3) the reflexive and transitive closure of the following *flow relation* F is a partial order $\preceq$ (called the *causal order*):

$$F = \{(u, v) : (pred(v))(u) = 1 \text{ or } (suc(u))(v) = 1\},$$

(2) $m : N_A \to N$ is a morphism from N_A to N,

(3) $i = (i(p) : p \in P_N)$ is an *arrangement of minimal elements*, where each $i(p)$ is an enumeration of the set of minimal $x \in X$ with $m(x) = p$,

(4) $j = (j(p) : p \in P_N)$ is an *arrangement of maximal elements*, where each $j(p)$ is an enumeration of the set of maximal $x \in X$ with $m(x) = p$,

where a process instance $A = (X, Y, pred, suc, m, i, j)$ is regarded to be equivalent to a process instance $A' = (X', Y', pred', suc', m', i', j')$ if there exists an isomorphism $f : N_A \to N_{A'}$ such that m is the composition of f and m', i is transformed into i', and j is transformed into j'.

For a process instance $A = (X, Y, pred, suc, m, i, j)$ we denote by U the set $X \cup Y$ and we use subscripts, U_A, X_A, Y_A, $pred_A$, suc_A, F_A, $\preceq_A$, m_A, i_A, j_A, when necessary.

Such a process instance A represents (a segment of) a concrete run of the system represented by the net N. Each $x \in X$ with $m(x) = p$, called in the sequel a *token*, represents a token residing in the place p. Each $y \in Y$ with $m(y) = t$, called in the sequel an *event*, represents an execution of the transition t. Each maximal antichain Z of U with respect to the causal order $\preceq$ such that $Z \subseteq X$, called a *cut* of A, represents a possible state of the process represented by A. In particular, the set of minimal elements of U represents the *initial state*, and the set of maximal elements represents the *final state*. The arrangements of minimal and maximal elements associate to such elements instance independent identifiers.

For each place p of N we have a *one-token* process, written as p, with a process net consisting of a single place element (token) x such that $m(x) = p$. For each transition t of N we have a *one-event* process, written as t, with a process net consisting of a single transition element (event) y such that $m(y) = t$ and of related place elements (tokens). We have also the *empty* process without tokens and events, nil.

Processes without events are called *process symmetries*. Symmetries whose arrangements of minimal elements are identical with arrangements of maximal elements are called *process identities* and they may be regarded as multisets of places.

By $Processes(N)$ we denote the set of processes of N.

In [Wnk 95] it has been observed that processes of a net N can be represented as isomorphism classes of structures which can be obtained from standard process instances by replacing the explicit representation of executions of transitions by weights between elements representing tokens, where for each pair of elements the respective weight specifies the sequences of transitions corresponding to all the possible maximal chains from the first element of the pair to the second element. Consequently, to each process α a table of weights between minimal and maximal elements of an instance of this process, $table(\alpha)$, can be associated, where each weight is an element of $(T_N^*)^{\oplus}$, the free semiring (with a zero element $\perp$ and a unit element ε) generated by the set T_N.

In the present formulation $table(\alpha)$ can be defined as the function which associates to each pair (p, q) of places of N a matrix $(table(\alpha))(p, q)$ with elements given for each instance A of α by the formula

$$((table(\alpha))(p, q))(r, s) = \Sigma(m(c) : c \in Maxchains((p, r), (q, s))),$$

where

(1) $1 \le r \le length(i_A(p))$ with $length(i_A(p))$ denoting the length of the sequence $i_A(p)$ of minimal elements of A with the label p,
(2) $1 \le s \le length(j_A(q))$ with $length(j_A(q))$ denoting the length of the sequence $j_A(q)$ of maximal elements of A with the label q,
(3) $Maxchains((p, r), (q, s))$ denotes the set of maximal chains from $(i_A(p))(r)$, the r-th element of $i_A(p)$, to $(j_A(q))(s)$, the s-th element of $j_A(q)$,
(4) $m(c)$ denotes the string $m(y_1)...m(y_k)$ of transitions of N for each maximal chain $c = (x_0 \preceq_A y_1 \preceq_A x_1 \preceq_A ... \preceq_A y_k \preceq_A x_k)$ from $x_0 = (i_A(p))(r)$ to $x_k = (j_A(q))(s)$,
(5) the result of summation on the right hand side is defined as the zero element $\perp$ if the respective set of maximal chains is empty.

In particular, for each one-token process corresponding to a place p we have a table whose the only nonempty matrix is $(table(p))(p, p)$, this nonempty matrix has exactly one element, and this element is the unit element ε, for each one-event process corresponding to a transition t we have a table whose all nonempty matrices have all elements equal to the one-element string t, and for the empty process nil we have the table consisting of empty matrices, nil'.

Tables corresponding to processes can be regarded as *partitioned matrices*.

Formally, a partitioned matrix of elements of a semiring S with component matrices indexed by pairs of elements of a set V, or briefly a V-matrix over S, is a mapping A from $V \times V$ to the set of matrices over S such that $height(A(v, v')) = height(A(v, v''))$ and $width(A(v', v)) = width(A(v'', v))$ for all $v, v', v'' \in V$, where $height(M)$ and $width(M)$

denote respectively the height (= the number of rows) and the width (= the number of columns) of a matrix M.

For $v \in V$ by $height(A(v,.))$ we denote the common value of the quantities $height(A(v,v'))$, and by $width(A(.,v))$ we denote the common value of the quantities $width(A(v',v))$. For $v \in V$ and $1 \leq r \leq height(A(v,.))$ by $(A(v,.))(r,.)$ we denote the family $((A(v,v'))(r,s) : v' \in V, 1 \leq s \leq width(A(.,v')))$ and we call such a family a *row* of A of type v. Similarly, by $(A(.,v))(.,s)$ we denote the family $((A(v',v))(r,s) : v' \in V, 1 \leq s \leq hight(A(v',.)))$ and we call such a family a *column* of A of type v.

If $height(A(v,.)) = width(A(.,v))$ for all v and there exists a permutation f_v of the sequence $1,...,height(A(v,.))$ such that $(A(v,v))(r,s)$ is different from $\perp$, the zero element of S, only for $s = f_v(r)$, and then it coincides with ε, the unit element of S, and if each $A(v,v')$ with $v \neq v'$ is a zero matrix, then we call A a *matrix symmetry.* If also all f_v are identity permutations then we call A a *matrix identity.*

By $Pmatrices(V,S)$ we denote the set of V-matrices over S.

For a net N the correspondence $\alpha \mapsto table(\alpha)$ is a mapping from the set of processes of N, $Processes(N)$, to the set of P_N-matrices over $(T_N^*)^{\oplus}$, $Pmatrices(P_N,(T_N^*)^{\oplus})$.

In this framework our problem can be formulated as the problem of characterizing those P_N-matrices over $(T_N^*)^{\oplus}$ which belong to the image of the set $Processes(N)$ under the mapping $\alpha \mapsto table(\alpha)$, and of characterizing the equivalence induced by this mapping in the set $Processes(N)$.

3. Algebraic framework

The solution which we present in this paper is based on the fact that both processes of a net N and the corresponding partitioned matrices can be obtained by combining symmetries and one-token- and one-event processes of N and the corresponding partitioned matrices.

Processes of a net N can be combined with the aid of operations which can be defined as follows.

For each process α there exists a unique identity process $\partial_0(\alpha)$, called the *source* of α (resp.: a unique identity process $\partial_1(\alpha)$, called the *target* of α), whose each instance can be obtained from an instance A of α by restricting A to the set of minimal (resp.: maximal) elements of U_A and by replacing j_A by i_A (resp.: i_A by j_A). The correspondences $\alpha \mapsto \partial_0(\alpha)$ and $\alpha \mapsto \partial_1(\alpha)$ are operations such that

$$\partial_0(\partial_0(\alpha)) = \partial_1(\partial_0(\alpha)) = \partial_0(\alpha),$$
$$\partial_0(\partial_1(\alpha)) = \partial_1(\partial_1(\alpha)) = \partial_1(\alpha).$$

A process γ is said to *consist* of a process α *followed* by a process β if each its instance C has a cut Z and an arrangement of elements of Z into a family $r = (r(p) : p \in P_N)$ of enumerations of the sets $m_C^{-1}(p) \cap Z$ such that:

(1) the restriction of C to the set $\{u \in U_C : u \preceq_C z$ for some $z \in Z\}$ with r playing the role of arrangement of maximal elements, $head_{Z,r}(C)$, is an instance of α,
(2) the restriction of C to the set $\{u \in U_C : z \preceq_C u$ for some $z \in Z\}$ with r playing the role of arrangement of minimal elements, $tail_{Z,r}(C)$, is an instance of β.

For every two processes α and β such that $\partial_1(\alpha) = \partial_0(\beta)$ there exists a unique process $\alpha;\beta$ that consists of α followed by β. The correspondence $(\alpha,\beta) \mapsto \alpha;\beta$ is an associative partial operation such that

$$\partial_0(\alpha;\beta) = \partial_0(\alpha),$$
$$\partial_1(\alpha;\beta) = \partial_1(\beta),$$
$$\partial_0(\alpha);\alpha = \alpha;\partial_1(\alpha) = \alpha.$$

We call it a *sequential composition.*

A process γ is said to *consist* of a process α *accompanied* by a process β if each its instance C has a partition $p = (U', U'')$ of U_C into two disjoint subsets U',U'' such that:

(1) u',u'' are incomparable whenever $u' \in U'$ and $u'' \in U''$,
(2) each $i_C(p)$ is $(i_C(p)|U')(i_C(p)|U'')$, the concatenation of the restrictions of $i_C(p)$ to U' and U'',
(3) each $j_C(p)$ is $(j_C(p)|U')(j_C(p)|U'')$, the concatenation of the restrictions of $j_C(p)$ to U' and U'',
(4) the restriction of C to U' with the arrangement of minimal elements given by $i_C|U' = (i_C(p)|U' : p \in P_N)$ and the arrangement of maximal elements given by $j_C|U' = (j_C(p)|U' : p \in P_N)$, $left_p(C)$, is an instance of α,
(5) the restriction of C to U'' with the arrangement of minimal elements given by $i_C|U'' = (i_C(p)|U'' : p \in P_N)$ and the arrangement of maximal elements given by $j_C|U'' = (j_C(p)|U'' : p \in P_N)$, $right_p(C)$, is an instance of β.

For arbitrary two processes α and β there exists a unique process $\alpha \otimes \beta$ that consists of α accompanied by β. The correspondence $(\alpha, \beta) \mapsto \alpha \otimes \beta$ is an associative (but not commutative) operation such that

$$\partial_0(\alpha \otimes \beta) = \partial_0(\alpha) \otimes \partial_0(\beta) (= \partial_0(\beta) \otimes \partial_0(\alpha)),$$
$$\partial_1(\alpha \otimes \beta) = \partial_1(\alpha) \otimes \partial_1(\beta) (= \partial_1(\beta) \otimes \partial_1(\alpha)),$$
$$(\alpha; \beta) \otimes (\gamma; \delta) = (\alpha \otimes \gamma); (\beta \otimes \delta) \text{ whenever } \alpha; \beta \text{ and } \gamma; \delta \text{ are defined.}$$

We call it a *parallel composition.*

Finally, for arbitrary identities (multisets) a and b there exists a unique symmetry $I_*(a, b)$ whose each instance C can be partitioned into an instance A of a and an instance B of b in the sense that $U_C = U_A \cup U_B$ with $U_A \cap U_B = \emptyset$, each $i_C(p)$ is $i_A(p)i_B(p)$, the concatenation of $i_A(p)$ and $i_B(p)$, and each $j_C(p)$ is $j_B(p)j_A(p)$, the concatenation of $j_B(p)$ and $j_A(p)$. The correspondence $(a, b) \mapsto I_*(a, b)$ is a partial operation such that

$$I_*(a, b); I_*(b, a) = a \otimes b,$$
$$(I_*(a, b) \otimes c); (b \otimes I_*(a, c) = I_*(a, b \otimes c),$$
$$I_*(\partial_0(\alpha), \partial_0(\beta)); (\beta \otimes \alpha) = (\alpha \otimes \beta); I_*(\partial_1(\alpha), \partial_1(\beta)).$$

We call it an *interchange.*

When equipped with the above operations the set of processes of N becomes a partial *algebra of processes* of N, $PROCESSES(N)$.

V-matrices over a semiring S can be combined with the aid of operations similar to those on processes of a net.

For each V-matrix A we have two identity V-matrices: a *source* $\partial_0'(A)$ and a *target* $\partial_1'(A)$, where $(\partial_0'(A))(v, v)$ is the identity matrix with $height(A(v, .))$ rows and $height(A(v, .))$ columns, and where $(\partial_0'(A))(v, v')$ with $v \neq v'$ is the zero matrix with $height(A(v, .))$ rows and $height(A(v', .))$ columns, $(\partial_1'(A))(v, v)$ is the identity matrix with $width(A(., v))$ rows and $width(A(., v))$ columns, and where $(\partial_1'(A))(v, v')$ with $v \neq v'$ is the zero matrix with $width(A(., v))$ rows and $width(A(., v'))$ columns.

For V-matrices A and B such that $\partial_1'(A) = \partial_0'(B)$ we have a unique V-matrix $A;' B$, where

$$(A;' B)(v, v') = \Sigma(A(v, v'')B(v'', v') : v'' \in V)$$

for all $v, v' \in V$, that is

$$((A;' B)(v, v'))(r, s) =$$
$$\Sigma((A(v, v''))(r, k)(B(v'', v'))(k, s) : v'' \in V, k \in \{1, ..., width(A(., v''))\})$$

for all the respective v, v', r, s.

Thus we have a partial operation $(A, B) \mapsto A;' B$. We call it a *multiplication* or a *sequential composition*.

For arbitrary V-matrices A and B we have a unique V-matrix $A \otimes' B$, where each $(A \otimes' B)(v, v')$ denotes the matrix of the form

$$\begin{bmatrix} A(v,v') & zero\ matrix \\ zero\ matrix & B(v,v') \end{bmatrix}$$

Thus we have an operation $(A, B) \mapsto A \otimes' B$. We call it a *juxtaposition* or a *parallel composition*.

For arbitrary identity V-matrices A and B we have a unique symmetry V-matrix $I'_*(A, B)$, where each $(I'_*(A, B))(v, v')$ denotes the matrix of the form

$$\begin{bmatrix} zero\ matrix & A(v,v') \\ B(v,v') & zero\ matrix \end{bmatrix}$$

Thus we have a partial operation $(A, B) \mapsto I'_*(A, B)$. We call it an *interchange*.

The operations just introduced on V-matrices enjoy all the properties of the corresponding operations on processes.

When endowed with these operations the set of V-matrices over S becomes a partial *algebra of V-matrices over S, $PMATRICES(V, S)$*.

From the respective definitions it follows that the correspondence $\alpha \mapsto table(\alpha)$ between processes of a net N and P_N-matrices over $(T_N^*)^{\oplus}$ is a homomorphism from the algebra $PROCESSES(N)$ to the algebra $PMATRICES(P_N, (T_N^*)^{\oplus})$ (cf. [Wnk 95]).

4. Solution

In [DMM 89] it has been shown that each processes α of a net N can be obtained by combining symmetries and one-token- and one-event processes of N in the sense that it can be represented in a *sequential form*

$$\sigma_1; \alpha_1; \ldots; \sigma_n; \alpha_n; \sigma_{n+1},$$

where σ_1,...,σ_n,σ_{n+1} are symmetries and α_1,...,α_n are processes of the form

$$\alpha_i = t_i \otimes p_{i1} \otimes \ldots \otimes p_{ik(i)}$$

with t_i denoting the one-event process corresponding to a transition t_i and p_{i1},...,$p_{ik(i)}$ denoting the one-token processes corresponding to places p_{i1},...,$p_{ik(i)}$. Such a representation, in general not unique, can be obtained by considering a chain of subsequent cuts of an instance of the considered process α (cf. [Wnk 95]).

As the correspondence $\alpha \mapsto table(\alpha)$ is a homomorphism, for each representation as above we have the corresponding representation

$$table(\alpha) = table(\sigma_1);' table(\alpha_1);' \ldots;' table(\sigma_n);' table(\alpha_n);' table(\sigma_{n+1}),$$

where $table(\sigma_1)$,...,$table(\sigma_n)$,$table(\sigma_{n+1})$ are symmetries and $table(\alpha_1)$,...,$table(\alpha_n)$ are partitioned matrices of the form

$$table(\alpha_i) = table(t_i) \otimes' table(p_{i1}) \otimes' \ldots \otimes' table(p_{ik(i)}).$$

Consequently, each partitioned matrix which corresponds to a process α of N can be decomposed into partitioned matrices corresponding to symmetries and to one-token- and one-event processes of N.

Also the converse is true. If a partitioned matrix A can be decomposed into partitioned matrices corresponding to symmetries and to one-token- and one-event processes of N then, due to the properties of operations on partitioned matrices and of the correspondence $\alpha \mapsto table(\alpha)$, it can be represented in the form

$$A = S_1 ;' A_1 ;' \dots ;' S_n ;' A_n ;' S_{n+1},$$

where $S_1,\dots,S_n,S_{n+1}$ are symmetries with $S_i = table(\sigma_i)$ for some process symmetries σ_i, and $A_1,\dots,A_n$ are partitioned matrices of the form

$$A_i = B_i \otimes' C_{i1} \otimes' \dots \otimes' C_{ik(i)}$$

with $B_i = table(t_i)$ for one-event processes t_i and $C_{i1} = table(p_{i1}),\dots,C_{ik(i)} = table(p_{ik(i)})$ for one-token processes $p_{i1},\dots,p_{ik(i)}$.

Consequently, $A_i = table(\alpha_i)$, where $\alpha_i = t_i \otimes p_{i1} \otimes \dots \otimes p_{ik(i)}$. As for identity processes the equality $\alpha = \beta$ is equivalent to the equality $table(\alpha) = table(\beta)$, we can compose sequentially processes $\sigma_1,\alpha_1,\dots,\sigma_n,\alpha_n,\sigma_{n+1}$, which gives a process α such that $table(\alpha) = A$.

Thus we obtain the following characterization of partitioned matrices corresponding to processes of N.

Theorem 1. A partitioned matrix $A \in Pmatrices(P_N, (T_N^*)^{\oplus})$ corresponds to a process α of a net N in the sense that $A = table(\alpha)$ iff it can be decomposed into symmetries and partitioned matrices corresponding to one-token- and one-event processes of N. $\square$

Let $Pmatproc(N)$ denote the set of partitioned matrices corresponding to processes of a net N, called *process matrices* of N. The stated theorem implies the following result.

Corollary. The set $Pmatproc(N)$ of process matrices of a net N is closed w.r. to the considered operations on partitioned matrices. Consequently, the restriction of the algebra $PMATRICES(P_N, (T_N^*)^{\oplus})$ to the subset $Pmatproc(N)$ of its carrier is a subalgebra, $PMATPROC(N)$, of this algebra. $\square$

The mapping α $\overset{t}{a}$ $ble(\alpha)$ is a homomorphism. It induces in the set $Processes(N)$ an equivalence. It suffices to define two processes α and β to be equivalent if they have identical sources, targets, and partitioned matrices.

This observation suggests that we might represent processes of a net by their partitioned matrices. Unfortunately, such a representation is not bijective, as it can be seen in figure 5.

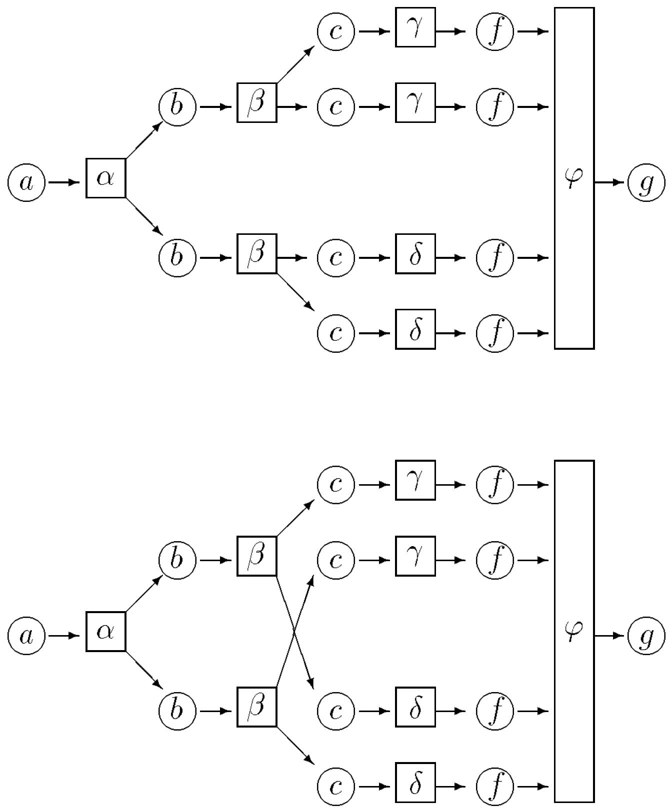

Figure 5

5. Concluding remarks

The representation of net processes by partitioned matrices allows one to answer various questions concerning such processes in purely algebraic ways.

In order to illustrate this let us consider two processes α and β of the net in figure 1, where α is the process shown in figure 6 and β is the process shown in figure 7, and suppose that we want to see if there exists (and to find) a process γ which consists of α and β in the sense that the token produced by α in C is consumed by β and the token produced by β in C is consumed by α.

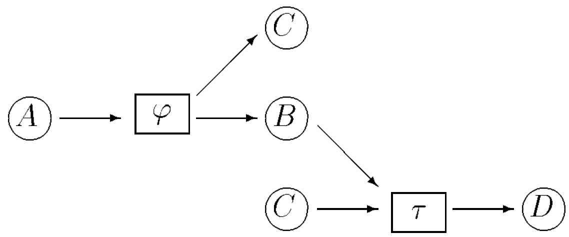

Figure 6

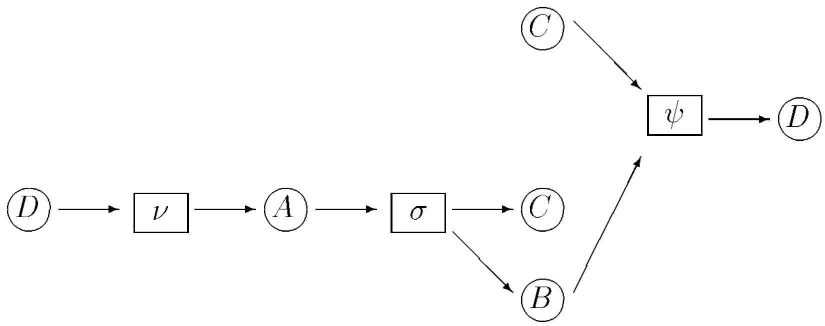

Figure 7

In the language of process matrices this problem reduces to the problem of finding a process matrix such that the system of equations in figure 8 is satisfied for all x,y,x',y' such that the systems of equations in figures 9 and 10 are satisfied for some z and t.

$$
\begin{array}{c|c|c}
 & A & D \\ \hline
 & x & y
\end{array}
\;;'\;
\begin{array}{c|cc}
 & D & D \\ \hline
A & s_{11} & s_{12} \\ \hline
D & s_{21} & s_{22}
\end{array}
=
\begin{array}{c|cc}
 & D & D \\ \hline
 & x' & y'
\end{array}
$$

Figure 8

$$
\begin{array}{c|c|c}
 & A & C \\ \hline
 & x & z
\end{array}
\;;'\;
\begin{array}{c|c|c}
 & C & D \\ \hline
A & \varphi & \varphi\tau \\ \hline
C & \perp & \tau
\end{array}
=
\begin{array}{c|c|c}
 & C & D \\ \hline
 & t & y'
\end{array}
$$

Figure 9

$$\begin{array}{c|c|c} & C & D \\ \hline & t & y \end{array} \;\;;'\;\; \begin{array}{c|c|c} & C & D \\ \hline C & \perp & \psi \\ \hline D & \nu\sigma & \nu\sigma\psi \end{array} \;=\; \begin{array}{c|c|c} & C & D \\ \hline & z & x' \end{array}$$

Figure 10

Equivalently, the required process matrix can be found as the matrix of the system of equations obtained from the systems in figures 9 and 10 by eliminating z and t.

By rewriting the two systems explicitly we obtain the following system of equations

$$\begin{gathered} x\varphi = t \\ x\varphi\tau + z\tau = y' \\ y\nu\sigma = z \\ t\psi + y\nu\sigma\psi = x'. \end{gathered}$$

By eliminating from this system z and t we obtain

$$\begin{gathered} x\varphi\psi + y\nu\sigma\psi = x' \\ x\varphi\tau + y\nu\sigma\tau = y'. \end{gathered}$$

Consequently, $s_{11} = \varphi\psi$, $s_{12} = \varphi\tau$, $s_{21} = \nu\sigma\psi$, $s_{22} = \nu\sigma\tau$.

Thus we obtain the partitioned matrix shown in figure 4.

In this case the partitioned matrix obtained as a solution corresponds to the process shown in figure 3.

In general, the existence of a suitable process may need a proof or must be checked by trying to decompose the respective partitioned matrix into symmetries and partitioned matrices corresponding to one-token- and one-event processes. At the moment we do not know any really simple algorithm of such a decomposition. However, often we do not need to decompose partitioned matrices of our concern since they are given by suitable expressions of the considered matrix calculus.

Concerning the representation of net processes by partitioned matrices it is worth to add that there are problems which become particularly simple when formulated with the aid of process matrices. Of this type are, for example, the problem of equality of processes represented by different expressions and the problem of evaluation of the execution time of a process.

References

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

[DMM 89] Degano, P., Meseguer, J., Montanari, U., *Axiomatizing Net Computations and Processes*, in the Proceedings of 4th LICS Symposium, IEEE, 1989, pp.175-185

[FMM 91] Ferrari, G.-L., Montanari, U., Mowbray, M., *Tracing Causality in Distributed Systems (Extended Abstract)*, in Proc. 3rd Workshop on Concurrency and Compositionality, Goslar, March 5-8, 1991, Hildesheimer Informatik - Bericht 6/91, Universität Hildesheim, pp.99-108

[GLT 80] Genrich, H.J., Lautenbach, K., Thiagarajan, P., *Elements of General Net Theory*, Net Theory and Applications (W. Brauer, Ed.), Springer LNCS 84, 1980, pp.21-163

[GR 83] Goltz, U., Reisig, W., *The Non-sequential Behaviour of Petri Nets*, Information and Control, vol.57, no.2-3, May-June 1983, pp.125-147

[GV 87] van Glabbeek, R., Vaandrager, F., *Petri Net Models for Algebraic Theories of Concurrency*, Proc. of PARLE Conference, Eindhoven, 1987, (J. W. de Bakker and A. J. Nijman, Eds.), Springer LNCS 259, pp.224-242

[MM 88] Meseguer, J., Montanari, U., *Petri Nets Are Monoids: A New Algebraic Foundation for Net Theory*, Proc. LICS, 1988, pp.175-185

[R 85] Reisig, W., *Petri Nets*, Springer, 1985

[Wnk 95] Winkowski, J., *Concatenable Weighted Pomsets and Their Applications to Modelling Processes of Petri Nets*, Institute of Computer Science of the Polish Academy of Sciences, Technical Report 784, August 1995, Fundamenta Informaticae, vol.28, no.3,4 (December 1996), pp.403-421

[Wnk 96] Winkowski, J., *A Representation of Processes of Petri Nets by Matrices*, Institute of Computer Science of the Polish Academy of Sciences, Technical Report 807, July 1996

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